UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE FÍSICA

Aprendizagem de conceitos não-estacionários por meio de redes neurais artificiais.

Evaldo Araújo de Oliveira Filho

Tese apresentada ao Instituto de Física da Universidade de São Paulo para a obtenção do título de Doutor em Ciências.

Orientador : Prof. Dr. Nestor Caticha

Banca Examinadora:

Prof. Dr. Carlos Eugênio Imbassahy Carneiro (IF-USP)

Prof. Dr. José Roberto Castilho Piqueira (EP-USP)

Prof. Dr. Nestor Caticha (IF-USP)

Prof. Dr. Osame Kinouchi Filho (FFCLRP-USP)

Prof. Dr. Silvio Roberto de Azevedo Salinas (IF-USP)

SÃO PAULO

2005

Prof. Armando Corbani Ferraz
Presidente da Comissão de Pós Graduação

FICHA CATALOGRÁFICA

Preparada pelo Serviço de Biblioteca e Informação do Instituto de Física da Universidade de São Paulo

Oliveira Filho, Evaldo Araújo de

Aprendizagem de conceitos não-estacionários por meio de redes neurais artificiais.

São Paulo, 2005.

(Doutoramento) - Universidade de São Paulo.
Instituto de Física - Departamento de Física Geral
Orientador: Prof. Dr. Nestor Caticha
Área de Concentração: Física

Unitermos:

1. Redes neurais artificiais;
2. Inferência bayesiana;
3. Mecânica estatística.

USP/IF/SBI-052/2005

Aos meus pais e irmãos.

Agradecimentos

Certamente houve muitos que me ajudaram de alguma forma durante o período de realização do doutorado (principalmente os nossos funcionários), por isso, mesmo não podendo entrar em detalhes, gostaria de agradecer inicialmente a esses colegas, aqui anônimos.

Essa pesquisa foi financiada pela Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) sob a orientação do Prof. Nestor Caticha, portanto, agradeço a ambos pela oportunidade concedida. Gostaria também de enfatizar a amizade oferecida pelo Prof. Nestor e sua compreensão nos momentos de dificuldades, principalmente após o término da bolsa de estudo. Também merece citação o Prof. Manfred Opper, da Aston University, que me recebeu e orientou por três meses em Birmingham.

Durante os três meses em Birmingham, contei com a ajuda e colaboração de Roberto Alamino, que se estendem até os dias de hoje. Ele e Fabiano Ribeiro foram meus principais parceiros do DFGE em conversas sobre nossas pesquisas, filosofias e vidas. Agradeço aos colegas de sala e departamento pelo companheirismo e em especial a Marcus Ferraz por me conceder a preferência no computador *banach*, após a *morte* do *bayes1*, computador que eu utilizava em meu trabalho. Esse ajuda foi essencial para o término da tese. Agradeço também a Beatriz Schmitz pelas dicas da língua portuguesa e por sua amizade.

Uma tese de doutorado não se faz apenas no laboratório ou sala de trabalho, ainda mais se acreditarmos que não se trata apenas da redação de um texto científico, mas da formação de um cientista. Assim, agradeço aos queridos amigos Ivan e Ise, pelo carinho e agradável convivência em nossa república; a Timóteo e aos irmãos da *University Bible Fellowship* e da *Missão Horizontes* pelo acompanhamento e orientação em grande parte do período referido; a Libério e aos irmãos do grupo *Cristãos USP* com quem tive inúmeras aventuras e aprendizados; a Raul Nogueira e sua esposa Graça pela orientação e cuidado pessoal e principalmente a minha mãe por todo amor e cuidado a mim dirigidos nesse longo tempo longe de casa.

Por último enfatizo minha dedicatória à minha família, que tanto amo, e meus agradecimentos a Deus por ser um suporte em minha vida.

Resumo

Num sentido geral, qualquer sistema (natural ou artificial) que incorpore informação contida numa amostragem de dados realiza aprendizagem. Dado um conjunto D de amostras que carrega informação sobre sua fonte geradora, existem diferentes medidas para quantificar a aprendizagem sobre ela e, portanto, uma boa representação de tal fonte. Contudo, não estamos interessados numa aprendizagem que apenas torne possível a reprodução de D por um sistema aprendiz, mas principalmente numa que torne possível a geração de novos dados condizentes com a fonte geradora. Portanto, uma vez fixado um sistema (máquina ou algoritmo), aprender significa encontrar um estado do sistema aprendiz que generalize a fonte geradora de D.

Em Mecânica Estatística as informações relevantes sobre os estados de qualquer sistema estão contidas em sua função de partição $\mathcal{Z}$. Logo, a inferência de qualquer variável é obtida tratando-se $\mathcal{Z}$, de forma que o seu conhecimento (cálculo) representa o conhecimento dos estados do sistema, ou seja, do próprio sistema. Num problema de aprendizagem bayesiana a função de partição é representada pela distribuição posterior a D (que já tenha incorporado as informações dos exemplos), $P(\cdot|D)$, obtida por meio da regra de Bayes $P(A,B) = P(A|B)P(B)$. Embora a abordagem bayesiana se enquadre originalmente em modelos da Mecânica Estatística em equilíbrio, sua utilização tem sido promissora também em cenários que podem ser interpretados como modelos de mecânica estatística fora do equilíbrio termodinâmico, sendo a aprendizagem de conceitos que mudam no decorrer do processo de aprendizagem um desses problemas que têm atraído bastante atenção.

O principal objetivo desta tese foi o estudo da aprendizagem bayesiana quando além do acesso ao conjunto D temos também a informação de que a fonte geradora de D é não-estacionária, introduzindo assim *tempo* num problema que de outra forma seria classificado como em equilíbrio. Em particular, estudamos a aprendizagem de conceitos com várias formas de dependência temporal por redes neurais (mais especificadamente, perceptrons), para a qual não é necessário modificar a verossimilhança do modelo. Assim nos concentramos na modificação do conhecimento *a priori* de forma a refletir a possibilidade de envelhecimento dos dados, numa es-

cala de tempo desconhecida. Ao introduzirmos uma distribuição de probabilidades priori para essa escala de tempo, nós encontramos uma distribuição posterior efetiva com uma cauda de decaimento algébrico que resultou num novo algoritmo com uma capacidade de adaptação satisfatória. Também aplicamos esse novo algoritmo na aprendizagem com ruído e discutimos algumas novas possibilidades sobre algoritmos para perceptrons.

Abstract

In a general sense, any system which incorporates knowledge from sample data can be called a learning machine (natural or artificial). Given a set D of samples which carries information about a rule, there are different measures of how much a system has learnt about the rule and therefore comprises a good representation of its. We are not only interested in learning that can reproduce D, but also generate new consistent data. Therefore, once fixed a system (a machine or an algorithm), to learn means to find a state of the system that generalizes the source rule of D.

We looked at Bayesian formulations of the learning problem, which is a formalism identical to Statistical Mechanics. Relevant knowledge about a given system is encoded in a partition function $\mathcal{Z}$. Then, any inference can be made by treating $\mathcal{Z}$, and if we know $\mathcal{Z}$ we know the system's properties. The function $\mathcal{Z}$ is the posterior distribution $P(\cdot|D)$ in the Bayesian approach, calculated by the Bayes' rule $P(A,B) = P(A|B)P(B)$. Although the Bayesian theory is naturally paralleled in equilibrium Statistical Mechanics, it holds the promise of leading to results in problems that can be classified as non-equilibrium. One of this problems that has been the subject of increasing attention is that of learning non-stationary concepts.

The aim of this thesis was to study Bayesian learning when in addition to the knowledge to the data set D we have the information that the rule which gave rise to the samples is non-stationary, thereby introducing *time* into what would otherwise, have been an equilibrium problem. In particular we studied learning of several forms of time dependent concepts by neural networks (more specifically, perceptrons), for which there is no need to change the likelihood. We concentrated on changing the prior knowledge in a way that reflects the aging possibility of the data on an unknown time scale. By introducing a prior probability distribution for the time scale, we found a effective posterior distribution with an algebraic decaying tail, which resulted in a new algorithm that was able to adapt satisfactory. We also applied the new algorithm to the learning with noise data and discussed some new possibilities about algorithms for perceptrons.

Sumário

Lista de Figuras

Prefácio

O processo de aprendizagem numa pesquisa científica nem sempre ocorre de maneira programada, ou digamos, linear. Muitas vezes não se tem conhecimento das características mais relevantes, ou interessantes, de um determinado problema no início do processo, o que dificulta grandemente a elaboração de perguntas chaves.

A idéia de estudar a aprendizagem de conceitos não-estacionários por redes neurais[1] não estava nos objetivos iniciais do nosso projeto de pesquisa. Inicialmente estudamos a relação entre a aprendizagem *on-line* e a *off-line* por meio de uma expansão generalizada tipo Gram-Charlier da distribuição posterior dos pesos sinápticos da rede [de Oliveira 2000]. O objetivo era traçar um caminho entre essas duas abordagens, em termos dos cumulantes da distribuição posterior. Como a utilização de cumulantes de ordem superior a dois não trazia diferenças significativas nas grandezas de interesse — por exemplo, erro de generalização — resolvemos procurar um cenário no qual essa diferença se tornasse significante. Por fim, ao não encontrarmos tal cenário ficamos apenas com a pergunta: *por que os dois primeiros cumulantes são suficientes para o perceptron?* e seguimos no estudo das *máquinas de vetores de suporte* e *processos gaussianos* [Scholkopf et al 2001]. Contudo, ao contrário das redes neurais artificiais, a construção de algoritmos on-line para as máquinas de vetores de suporte esbarrava em grandes dificuldades matemáticas, muito maiores que as encontradas pelos algoritmos off-line, que praticamente se resumiam na inversão de uma matriz.

Na procura de respostas para os problemas programados, esbarramos com o problema de conceitos não-estacionários e na pergunta que consideramos chave para esta tese: *como inserir a informação de envelhecimento dos conceitos a serem aprendidos?* Assim, caminhamos para o que eu considero ser o mais relevante da pesquisa realizada: o estudo da teoria bayesiana das probabilidades [Jaynes 2003, Bernado et al 2000] — embora ainda deva me confessar como um mero amador.

A teoria bayesiana das probabilidades recebeu seu nome em homenagem ao pastor presbiteriano e matemático Thomas Bayes, em conseqüência do trabalho "An essay towards solving a problem in the doctrine of chances", publicado após sua

[1] Na verdade, nos limitamos apenas ao caso de uma rede com um único neurônio, o *perceptron*.

morte por Richard Price em 1763 (*Phil. Trans. Roy. Soc.* 53, 370-418). Motivado pelo cálculo de uma distribuição para os parâmetros de uma distribuição binomial, Bayes anunciou o que é hoje conhecido como "teorema de Bayes" como uma das proposições necessárias:

> *If there be two subsequent events, the probability of the second b/N and the probability of both together P/N, and it being first discovered that the second event has also happened, the probability I am right [i.e., the conditional probability of the first event being true given that the second has happened] is P/b.*
>
> *Thomas Bayes*

Em outras palavras, a probabilidade de um evento A acontecer dado que foi observado o evento B é igual a razão $P(A,B)/P(B)$, logo, $P(A,B) = P(A|B)P(B)$. Assim, podemos dizer que esta tese é um pequeno fruto dessa idéia aparentemente simples aplicada no problema de aprendizagem em redes neurais artificiais.

No capítulo um apresentamos um ensaio sobre a evolução das idéias e estudos sobre o raciocínio humano, evidenciando dessa forma a natureza da teoria bayesiana. Em seguida, no capítulo dois, definimos o problema de aprendizagem e introduzimos o método utilizado para a construção de algoritmos bayesianos on-line. No capítulo três aplicamos os algoritmos obtidos no capítulo anterior, na aprendizagem de conceitos não-estacionários. Por fim, apresentamos um novo algoritmo no capítulo quatro, aplicando-o nos mesmos casos anteriores. Concluímos o trabalho no capítulo cinco, aonde também apresentamos algumas perspectivas.

Enfatizamos que utilizamos apenas *software livre* na elaboração desta tese. O sistema operacional utilizado foi *Debian* (Linux); o texto foi escrito na linguagem LaTeX 2_ε por meio do editor *Emacs*; as figuras foram desenhadas pelo *Xfig*; as simulações e cálculos numéricos foram escritas em *Fortran* e executados por meio do compilador *g77* e os gráficos foram desenhados no *Grace*. Esses programas podem ser encontrados no sítio http://www.debian.org.br/ ou no http://www.gnu.org/.

Notação

Abreviações

ABOff	Algoritmo bayesiano off-line
ABOn	Algoritmo bayesiano on-line
ABOn EA	Algoritmo bayesiano on-line escalar adaptativo
ABOn escalar	Algoritmo bayesiano on-line escalar
ABOn tensorial	Algoritmo bayesiano on-line tensorial
AO	Algoritmo ótimo
DG	(Algoritmo tipo) descida pelo gradiente
GN	(Algoritmo do) gradiente natural
ME	Maximização da entropia
RNA	Rede neural artificial
RT	Reguladores de Tikhonov

Símbolos

$\boldsymbol{\omega}$	Vetor sináptico — $\boldsymbol{\omega} \in \Re^N$.
$\bar{\boldsymbol{\omega}}$	Vetor sináptico *professor* — $\boldsymbol{\omega} \in \Re^N$.
$\hat{\boldsymbol{\omega}}$	Estimativa de $\bar{\boldsymbol{\omega}}$.
ζ	Variância sináptica — $\zeta \in \Re$.
C	Covariância sináptica — $C \in \Re^N \times \Re^N$.
$\boldsymbol{\xi}$	Vetor estímulo — $\boldsymbol{\xi} \in \Re^N$.
$v = \boldsymbol{\omega} \cdot \boldsymbol{\xi}$	Produto escalar — $v = \sum_{i=1}^{N} \omega^i \xi^i$.
$M = \boldsymbol{\xi} \otimes \boldsymbol{\xi}$	Produto tensorial — $M^{ij} = \xi^i \xi^j$.

τ	Campo sináptico — $\tau \propto \boldsymbol{\omega} \cdot \boldsymbol{\xi}$.
$\bar{\tau}$	Campo sináptico professor.
$\hat{\tau}$	Estimativa de $\bar{\tau}$.
σ	Reação a $\boldsymbol{\xi}$ — $\sigma \in \{-1; 1\}$.
ρ	Correlação entre $\bar{\boldsymbol{\omega}}$ e $\hat{\boldsymbol{\omega}}$.
α	Razão entre o número de exemplos fornecidos e a dimenção de $\boldsymbol{\omega}$.
$\langle ... \rangle_{\boldsymbol{u}}$	$\int \mathrm{d}^N u \; ... \; P(\boldsymbol{u})$.
D_{KL}	Divergência de Kullback-Leibler.
e_g	Erro de generalização.
e_p	Erro de predição.
e_r	Erro residual.

1

Introdução.

A busca pela compreensão do funcionamento da mente humana é quase tão antiga quanto a própria história da humanidade, tanto no sentido de alma, pensamento e emoções[1], quanto em relação a partes do corpo humano que estariam diretamente relacionados a tais, como sede.

Embora hoje seja bem conhecido que o cérebro é o órgão pelo qual podemos dizer que a mente age, nem sempre acreditou-se nisso. Por exemplo, os antigos egípcios criam que o coração era o cerne do homem, a essência da vida, bem como a fonte do bem ou mal. Tal pensamento era bastante popular na antiguidade[2] e até defendido por grandes nomes da ciência antiga. No entanto, é necessário esclarecer que provavelmente a palavra "coração" tenha sido anterior à descoberta do próprio órgão chamado *coração* (o coração físico) e referia-se a algo da essência interna do homem (intelecto, vontade e emoções). Como o coração físico é um órgão vital para a vida do homem, é perfeitamente compreensível que tenha sido apelidado por *coração*. O que afirmamos no início do parágrafo é que em várias sociedades, entre estas a egípcia, realmente se acreditava que o coração físico era a moradia da essência do homem.

A mudança do coração para o cérebro como o cerne do homem provavelmente começou no quinto século a.C.. Por volta de 450 a.C., um médico grego, conhecido por *Alcmaneon de Crotona*, baseando-se em seus conhecimentos sobre anatomia

[1] Ou estados da alma (alegria, tristeza, raiva,...).

[2] Aqui definimos por antiguidade o período da história humana anterior à vida de Jesus Cristo — como homem.

animal, concluiu que era o cérebro a sede dos pensamentos e emoções e não o coração. Ainda no mesmo século outros homens (Demócrito, Diógenes, Platão e Teófrasto) atribuíam ao cérebro o comando das atividades corporais, de maneira que cada vez mais não só se dava atenção ao cérebro como se colocava na posição de órgão mais importante do homem, detentor do seu cerne. Veja por exemplo a excelente declaração feita por Hipócrates (460-379 a.C.), onde ele exalta a função do cérebro no homem [Finger 1994]:

> *Deveria ser sabido que ele é a fonte do nosso prazer, alegria, riso e diversão, assim como nosso pesar, dor, ansiedade e lágrimas, e nenhum outro que não o cérebro. É especificamente o órgão que nos habilita a pensar, ver e ouvir, a distinguir o feio do belo, o mau do bom, o prazer do desprazer. É o cérebro também a sede da loucura e do delírio, dos medos e sustos que nos tomam, muitas vezes à noite, mas às vezes também de dia; é onde jaz a causa da insônia e do sonambulismo, dos pensamentos que não ocorrerão, deveres esquecidos e excentricidades.*
>
> *Hipócrates*

Como pode ser visto tal declaração é bem condizente, ou próxima, dos atributos do cérebro conhecidos hoje. No entanto, ainda havia resistências à retirada do coração como cerne, mesmo tendo a defesa de homens renomados. No quarto século a.C., Aristóteles contrariava tal idéia dizendo que o órgão dos pensamentos e emoções era o coração e que o cérebro era meramente um radiador designado para esfriar o coração, embora também afirmasse que o órgão do pensamento não era a base para o pensamento, pois tal base era imaterial e, portanto, não poderia ser encontrado em nenhum lugar do corpo humano.

Durante os séculos seguintes foram-se acumulando conhecimentos a respeito do coração e do cérebro e a resistência ao cérebro, sendo o cerne do homem, foi caindo lentamente.

No quarto século d.C., Nemésio, bispo de Emesia, relatou em seu livro *Da natureza do homem* que a alma não tinha uma sede, mas as funções da mente sim. Para tanto, entraram em ação os ventrículos cerebrais que seriam os responsáveis pelas operações mentais, desde a sensação até a memorização. Estes recebiam as informações vindas dos órgãos sensitivos e ali acontecia a análise sensorial. As imagens formadas eram levadas ao ventrículo médio, sede da razão, do pensamento e

do juízo. Depois entrava em ação o último ventrículo, sede da memória. Tais ventrículos eram associados a canais por onde circulavam espíritos e assim realizavam as funções do cérebro. Com isso, mais uma vez se diferenciava a alma do raciocínio do homem.

A idéia de espírito nos ventrículos permaneceu por muito tempo e em meados do século XVII d.C., Descartes, filósofo e matemático francês, propôs a idéia que o cérebro funcionava como uma máquina. Ele dizia que os nervos do homem eram cheios de espíritos de animais que levavam informações motoras e sensoriais para os ventrículos do cérebro, à semelhança dos fluidos hidráulicos numa máquina. Assim, Descartes descreveu o cérebro como sendo um máquina, trazendo com isso uma rica analogia apesar dos controversos espíritos. Embora o conceito do cérebro-máquina tenha permanecido até hoje, no século seguinte a teoria dos espíritos nos ventrículos foi *por água abaixo* com a demonstração da natureza elétrica na condução nervosa, publicada pelo fisiologista italiano Luigi Galvani (séc. XVIII d.C.).

Com o passar dos séculos, as descobertas sobre a fisiologia e funcionamento do cérebro acumularam-se a uma velocidade cada vez maior, acompanhando uma gradual dissociação entre o estudo do "cerne do homem" e o estudo do "raciocínio humano". Esse acréscimo na velocidade se deu devido ao desenvolvimento paralelo das teorias físicas e suas aplicações na construção de novos equipamentos de pesquisa. O microscópio, construído pela primeira vez em 1595 pelos holandeses Hans e Zacharias Jansen, foi um desses equipamentos fundamentais para o avanço das pesquisas sobre o cérebro e suas funções. Embora tenha sido muito rudimentar nos seus primeiros séculos de "vida", no século XIX d.C. o microscópio recebeu uma considerável melhoria na sua resolução, impulsionando novas descobertas sobre estruturas internas do cérebro. Uma conseqüência notável dessa melhoria foi a descrição das primeiras células neurais, ocorrendo esta na década seguinte à construção do primeiro microscópio acromático. Em conseqüência, os cientistas ficaram maravilhados com a rede fina e extremamente complexa de processos filamentares que parecia tomar por completo o sistema nervoso. Qual seria a função desses processos filamentares? Teria realmente alguma relação com o raciocínio[3]? Apesar dos cientistas não terem imediatamente tais respostas, eles tinham certeza da importância de tal rede para a compreensão do funcionamento do cérebro. Finalmente, em 1863, Otto Deiters obteve imagens claras e completas sobre os neurônios, isolando-os sob o microscópio.

[3]Dizemos, a rede de filamentos interna ao cérebro.

Daí surgiu o conhecimento da divisão estrutural do neurônio: *soma* (corpo celular), axônio e dendritos; reconhecida ainda nos dias de hoje.

Os avanços nos equipamentos e técnicas não pararam, de forma que no início do século XX d.C. já era bem aceito que todas as funções cerebrais eram resultantes da intensa transmissão de mensagens elétricas (e/ou neurotransmissores) pela gigantesca rede de neurônios do cérebro. Logo, o mais recente alvo era decifrar a linguagem dos *impulsos nervosos* que já se sabia serem "tudo ou nada", ou de uma forma mais ampla, lidar com a questão: *seria possível modelar o raciocínio humano?*

As teorias e ferramentas criadas para responder a pergunta do parágrafo anterior, na verdade, foram criadas e desenvolvidas muito antes da descoberta da rede de neurônios do cérebro. A discussão sobre o que seria o raciocínio e como funcionaria, ou como construí-lo, aparentemente seguiu o mesmo caminho greco-europeu. Como mencionamos, o estudo do raciocínio humano começou a se diferenciar do estudo do pensamento ou alma do homem no início do primeiro milênio d.C., o que, em minha opinião, contribuiu fortemente para o desenvolvimento da compreensão do raciocínio, escapando de questões intratáveis.

Apesar dos conceitos sobre mente, pensamento e raciocínio já terem sido bastante discutidos até o primeiro milênio d.C., a estruturação e elaboração destes numa forma mais precisa só foi realizada no século XVII d.C. com Descartes. Além de comparar o cérebro com uma máquina, Descartes também estruturou, ou melhor, organizou e elaborou, "o conhecimento do conhecimento humano" — como este funcionaria. Seus trabalhos sobre a estruturação do conhecimento tiveram um grande impacto na ciência, não só com relação à estrutura do raciocínio humano mas principalmente ao desenvolvimento do método da ciência. Além disso, Descartes também defendia que a matemática era uma linguagem comum da mente[4], princípio tal cada vez mais popular. Nessa direção, aproximadamente dois séculos depois, George Boole construiu uma álgebra binária[5] que representaria as "leis do pensamento"[6], revolucionando mais uma vez a teoria do raciocínio. Boole tinha uma clara compreensão da magnitude do seu trabalho, o que era evidenciado tanto por seu próprio título quanto pela própria elucidação do seu objetivo — veja a citação abaixo:

[4]Idéia originada na Grécia antiga.

[5]Hoje conhecida por álgebra booleana.

[6]Abreviação do título: *Uma Investigação das Leis do Pensamento, em que se Fundamentam as Teorias Matemáticas da Lógica e Probabilidades.* Trabalho publicado por G. Boole em 1854.

O motivo do presente tratado é investigar as leis fundamentais do funcionamento do cérebro através das quais o raciocínio se realiza; expressá-las através da linguagem do cálculo e, sobre este fundamento, estruturar a ciência da lógica e construir o seu método; fazer deste método a base de todos os métodos para aplicação da doutrina matemática das probabilidades; e, finalmente, recolher dos vários elementos verdadeiros trazidos para serem examinados no curso destas investigações alguma provável sugestão a respeito da natureza e constituição da mente humana.

G. Boole, 1854 - Leis do pensamento

No entanto, as *Leis do Pensamento* não foram bem recebidas, na verdade, podemos dizer que foram negligenciadas. A razão para tal não parece muito clara, mas talvez tenha ocorrido devido aos pesquisadores estarem procurando uma alternativa para a teoria das probabilidades apresentada no início do século XIX d.C.

Apesar da teoria das probabilidades já estar razoavelmente desenvolvida, principalmente devido a Bernoulli (séc. XVIII d.C.), Thomas Bayes (séc. XVIII d.C.) e posteriormente Laplace (séc. XIX d.C.), ainda existiam algumas lacunas entre as idéias conceituais das probabilidades e o que se esperava de uma teoria matemática "rigorosa". Na teoria proposta por Bayes em *An Essay Toward Solving a Problem in the Doctrine of Chances*, a probabilidade representava um "grau de crença", ou seja, o quanto se pensava ser alguma coisa verdadeira baseando-se na evidência disponível. Isto causava um certo desconforto para quantificar e operar crenças, ou seja, pensamentos. Apesar de Laplace ter desenvolvido analiticamente as idéias apresentadas por Bernoulli[7], reforçando o trabalho de Bayes, ainda havia muita desconfiança e indisposição por parte da maioria dos matemáticos e físicos (se não maioria, com certeza os mais influentes) para tal teoria. A principal crítica era que o conceito de probabilidade apresentado era muito vago e subjetivo para ser base de uma teoria matemática. Assim, no século XX d.C. houve um grande esforço para "consertar" as probabilidades e por fim acabaram definindo uma nova probabilidade que representava a freqüência relativa de um evento numa repetição de grandes conjuntos de amostragem. Com esse novo conceito de probabilidade, aparentemente ganhou-se mais objetividade na teoria, uma vez que "freqüências" podem ser medidas. No entanto, com isto limitou-se consideravelmente o espaço da validade ou aplicações das

[7]Expressadas no livro "A arte da conjectura", publicado por Bernoulli em 1713.

probabilidades.

Assim, mesmo com a visível riqueza da proposta bayesiana, bem como sua correspondência qualitativa com o senso comum, a abordagem freqüencista foi gradualmente ganhando espaço frente à bayesiana. Essa tendência era grandemente facilitada pela falta de argumentação matemática sobre a adoção das regras de inferência obtidas a partir da proposta de Bayes sob algum critério de otimização[8]. Já o trabalho de Boole não abordava diretamente essas questões na teoria das probabilidades, de forma a se manter neutro, embora viesse a ser fundamental para a construção dos processadores artificiais — que são utilizados nos nossos computadores.

O problema da abordagem freqüencista para a construção de processadores de informações é que nem sempre trabalhamos com a inferência de variáveis aleatórias, principalmente nas decisões tomadas no dia-a-dia. Posto isto, terminamos essa seção com a reapresentação da pergunta: seria possível construir uma teoria matemática para modelar o pensamento humano, ou melhor, *seria possível conciliar a teoria das probabilidades, inferência e lógica com o raciocínio humano?*

1.1 A resposta bayesiana.

O século XX d.C. começou então com o grande e emocionante desafio de construir uma teoria que funcionasse qualitativamente como o cérebro, ou seja, que processasse as informações numa maneira semelhante. Após praticamente um século de análise e discussão sobre o conjunto ou espaço de ferramentas adequadas para a construção de tal teoria, já tinha-se a convicção que a teoria das probabilidades era a resposta procurada, de maneira que bastava dar uma boa fundamentação matemática e conceitual, relacionando-a com a lógica simbólica.

Embora a linha ortodoxa tivesse grandes dificuldades em lidar conceitualmente com a inferência de variáveis que não fossem aleatórias, ela teve o seu auge com os trabalhos de Andrey Kolmogorov. O trabalho apresentado por Kolmogorov na verdade não precisava, ou partia, do conceito de freqüência de eventos, mas simplesmente dos trabalhos de Emile Borel sobre teoria da medida e dos conjuntos. Em seu

[8]Perceba que estamos falando das regras utilizadas num processo de inferência e não dos resultados obtidos. Até porque a boa qualidade da inferência bayesiana é bem conhecida desde a época de Laplace, quando ele estimou a massa de Saturno.

livro *Fundações do cálculo de probabilidades*[9], ele apresentou o sistema axiomático para a teoria das probabilidades baseado na teoria da medida e dos conjuntos, fundamentando matematicamente e definindo as regras necessárias para trabalhar-se com probabilidades. O impacto de tal trabalho foi tanto que logo se tornou referência e formulação básica para a teoria das probabilidades. Assim, aparentemente o problema estava resolvido. A partir de um conjunto de axiomas dado arbitrariamente, definiam-se as regras pelas quais as probabilidades deveriam ser tratadas (regra da soma, produto, ...), enquanto o tratamento lógico de proposições (as operações realizadas) era substituído pelas operações de conjunto no espaço dos pontos referentes às proposições. Porém, a dificuldade conceitual, apesar de atenuada, ainda persistia.

Alguns anos após o trabalho de Kolmogorov, Richard Cox [Cox 1946] demonstrou que também era possível construir uma teoria fundamentada matematicamente a partir da proposta de Laplace, obtendo-se as regras de Kolmogorov por conseqüência. Além disso, a arbitrariedade dos axiomas de Kolmogorov era substituída por critérios de consistência e qualitativa correspondência com o raciocínio humano [Jaynes 1988]. Mas será que era só isso? Apenas uma maneira alternativa de se reobter as regras de Kolmogorov? Não!

Desde a última década, observa-se um crescimento significativo do número de publicações de trabalhos que utilizam uma abordagem bayesiana. Isso não se deve propriamente a um modismo, mas aos ótimos resultados obtidos. Além disso, como a própria teoria bayesiana engloba a freqüencista, sempre teremos em geral uma qualidade no mínimo semelhante à obtida pela modelagem ortodoxa[10], quando esta é bem aplicada.

A teoria bayesiana traz consigo um conjunto de princípios de otimização que foram desenvolvidos e estabelecidos por caminhos independentes da teoria das probabilidades, mas que são cruciais para um bom tratamento de dados e modelagens em geral. Dentre esses princípios encontram-se a maximização da entropia, os reguladores de Tikhonov e o princípio de Ockham. A realização natural de tais princípios pela teoria bayesiana é um dos seus grandes trunfos. A seguir, discutiremos um pouco a idéia de cada um desses três princípios, apresentando suas relações com uma modelagem bayesiana. Para aqueles que desejarem conhecer mais sobre tais assuntos sugerimos as referências: [Jaynes 1982] para entropia máxima; [Kirsch 1996]

[9]Grundbegriffe der Wahrscheinlichkeitsrechnung, publicado por Kolmogorov em 1933 na Alemanha.

[10]Definimos como estatística e probabilidade ortodoxas aquelas oriundas da teoria freqüencista.

para reguladores e [Jefferys et al 1992] para o princípio de Ockham.

Entropia máxima. Do ponto de vista da "teoria da informação", a entropia representa a quantidade de informação necessária para sair de um estado de incompleta informação para outro de completo conhecimento, ou seja, uma quantidade de informação necessária para especificar por completo o estado no qual um determinado sistema se encontra. Nesse contexto, o princípio da maximização da entropia (ME) estabelece que inferências baseadas em dados incompletos (que não sejam suficientes para determinar unicamente a solução do problema) devem ser feitas por meio da distribuição de probabilidades que maximiza a entropia de acordo com os dados conhecidos, ou seja, satisfazendo os vínculos impostos pelo problema. Isso significa que, dentre todos os modelos possíveis, devemos escolher o que tiver maior entropia.

Considerando que o nosso conhecimento de um dado sistema seja representado por um conjunto de vínculos do tipo $\int \mathrm{d}^N\omega P(\omega)V(\omega) = A$ [11], o nosso problema consiste em achar um modelo $P(\omega)$ que satisfaça tais vínculos. Assim, sob o critério de maximização da entropia, a solução desejada é dada pela distribuição $P(\omega)$ que maximiza a lagrangeana:

$$\mathcal{L}(P(\cdot),\boldsymbol{\lambda}) = -\int \mathrm{d}^N\omega P(\omega)\ln P(\omega) - \sum_i \lambda_i \left(\int \mathrm{d}^N\omega V_i(\omega)P(\omega) - A_i\right), \quad (1.1)$$

ou seja, $P(\omega) = \exp\{-\sum_i \lambda_i V_i(\omega) - 1\}$ — os multiplicadores λ_i são determinados pela imposição $\int \mathrm{d}^N\omega V_i(\omega)P(\omega) = A_i$.

Consideremos agora que não só temos os valores esperados dos potenciais V_i, como também uma distribuição inicial para as variáveis ω^i: $Q(\omega)$. Isso implica que a entropia de Shannon, ou melhor, resultante dos axiomas de Shannon [Caticha et al 2004], torna-se

$$S(P|Q) = -\int \mathrm{d}^N\omega P(\omega)\ln\left(\frac{P(\omega)}{Q(\omega)}\right), \quad (1.2)$$

que também é conhecida como *entropia relativa* (a menos do sinal negativo).Em termos da teoria da informação podemos dizer que a entropia 1.2 é uma medida da quantidade de informação necessária para especificar o atual estado do sistema partindo de um conhecimento *a priori*, expressado por $Q(\omega)$.

[11] Ou $\sum_i p_i f(\omega_i) = a$ para o caso de variáveis discretas.

Substituindo $\int \mathrm{d}^N\omega P(\omega)\ln P(\omega)$ por 1.2 na lagrangeana 1.1, encontraremos que a solução que maximiza a entropia será agora dada por:

$$P(\omega) = \exp\{-\sum_i \lambda_i V_i(\omega) - 1\}Q(\omega). \tag{1.3}$$

Considerando $\exp\{-\sum_i \lambda_i V_i(\omega) - 1\}$ proporcional à verossimilhança dos dados fornecidos, teremos que a expressão 1.3 para a distribuição posterior $P(\omega)$ é justamente a regra de Bayes.

Reguladores de Tikhonov. Os reguladores de Tikhonov (RT) foram desenvolvidos dentro do contexto dos *problemas mal-formulados*[12], onde temos que escolher uma solução dentre tantas outras que satisfazem os dados experimentais (ou vínculos matemáticos).

Como pode-se perceber no tópico anterior, a idéia principal para resolver tais problemas é restringir a classe de soluções admissíveis, introduzindo informações *a priori* adequadamente.

Diferentemente da ME, onde o conhecimento *a priori* consiste na informação que soluções com grandes entropias têm mais chance de serem reais, no método dos RT a informação *a priori* é relacionado a aspectos geométricos da solução desejada (suavidade, ...). Assim, uma vez estabelecido o que se espera de antemão da solução, construímos o funcional

$$\mathcal{F}(\omega) = \{\text{Erro padrão}\} + \lambda\{\text{Inf. geométricas}\},$$

por meio do qual escolheremos uma solução, minimizando-o.

O primeiro termo do funcional $\mathcal{F}$ de Tikhonov mede o desvio entre a resposta desejada e a obtida pela solução candidata, enquanto o segundo representa um termo de punição da complexidade regulada por λ. Por sua vez, λ, que é definido como um número real positivo, controla a relação entre a suficiência dos dados D e a informação prévia I. No limite $\lambda \to 0$, temos que a solução procurada é totalmente determinada pelos exemplos fornecidos em D. Para $\lambda \to \infty$, temos que a restrição

[12]De acordo com Hadamard, um problema é classificado como mal-formulado se violar ao menos um dos seguintes itens: possui solução; a solução é única; a solução depende continuamente dos dados, ou seja, é uma função contínua. Assim, podemos perceber que os problemas de interesse da teoria da aprendizagem estatística (aprendizagem por máquinas ou inteligência artificial) são tipicamente uma classe de problemas mal-formulados.

imposta pelo operador descrito em I é por si só suficiente para determinar a solução, significando também que os dados são irrelevantes.

De maneira semelhante ao tópico anterior, ao construirmos uma distribuição de probabilidades proporcional à exponencial de $\mathcal{F}$, temos uma distribuição posterior bayesiana por meio da qual podemos fazer a inferência dos parâmetros em questão. Dessa forma, minimizar o funcional $\mathcal{F}$ de Tikhonov é agora equivalente a maximizar a densidade de probabilidade posterior às informações conhecidas. Além disso, abrimos grandes possibilidades para o tratamento contínuo de tais parâmetros por meio da regra de Bayes: $P(\omega|D, I, d, i) \propto P(\omega|D, I)p(d, i|\omega)$.

Perceba que, apesar das conexões dos métodos de ME e dos RT com o método bayesiano serem feitas por meio de uma simples aplicação da exponencial nos respectivos funcionais, seguido de uma identificação da distribuição *a priori*, isso é impossível para os métodos ortodoxos visto que em tal abordagem não tem sentido falar em distribuição *a priori*, mas apenas em verossimilhanças.

O princípio de Ockham. Originalmente anunciado por "William de Ockham"[13], o princípio estabelece que modelos desnecessariamente complexos não devem ser preferidos a modelos mais simples. Em outras palavras: quando temos duas explicações para um mesmo fenômeno, devemos escolher a mais simples.

Embora seja um princípio heurístico, ele reflete a posição usual e natural da grande parte dos cientistas. A melhor teoria para um fenômeno é a que reproduz o que conhecemos sobre ele, explicando-o na forma mais simples possível, até a chegada de nova informação.

Salientamos que o princípio de Ockham não refere-se à complexidade computacional, ou melhor, ao tempo de processamento de um algoritmo. Podemos dizer que está mais relacionado a plausibilidade do modelo (algoritmo). Por exemplo, se a verificação da falsidade de uma proposição A é mais fácil, ou acessível[14], que uma outra B e A se adequa aos dados D de forma equivalente a B, uma vez que não seja observada sua falsidade, A deve ser preferível a B.

A escolha entre diversos modelos M_i tanto na teoria ortodoxa como na bayesiana, é feita com base na *evidência* de cada modelo, definida como sendo a probabilidade de se observar D condicionada a M_i, ou seja, $P(D|M_i)$. No método da *Máxima*

[13]Teólogo e filósofo nascido em Ockham, Inglaterra, no final do século XIII.

[14]Para uma proposição ser falseável, em princípio tem que ser possível fazer uma observação ou experimento que mostre sua falsidade.

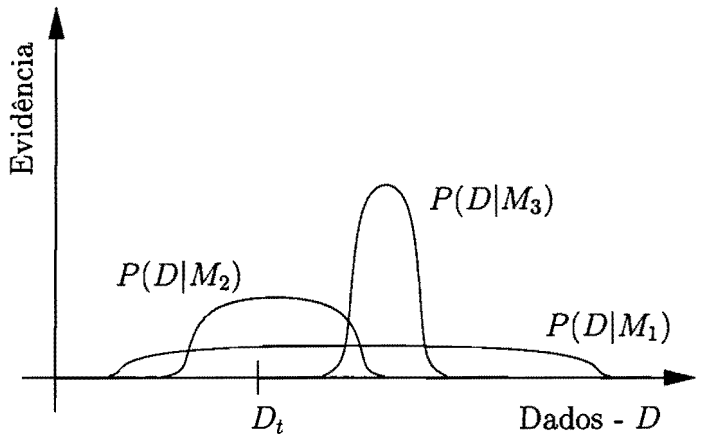

Figura 1.1: Evidência para três modelos diferentes.

Verossimilhança (MV), a evidência é obtida pelo seguinte esquema: dado um modelo M_i, encontram-se os parâmetros $\hat{\omega}$ de M_i que maximizam a verossimilhança dos dados de treinamento D; o modelo escolhido será aquele que tiver maior evidência $P(D|M_1) \equiv P(D|\hat{\omega}, M_1)$.

Na figura 1.1, temos a evidência para três modelos distintos entre si. O modelo M_1 é o mais expressivo, significando que variando os seus parâmetros é possível reproduzir uma grande quantidade de dados. Por sua vez, M_3 é o mais restritivo, enquanto que M_2 fica entre os dois outros. Para o conjunto D_t, temos que M_3 falha e M_1 e M_2 não. Como M_2 tem uma evidência para D_t maior que a de M_1, ele será o modelo escolhido.

No método bayesiano a evidência é calculada por meio da integral

$$P(D|M_i) = \int P(D|\omega, M_i)P(\omega|M_i)\mathrm{d}\omega.$$

Geralmente a posterior $P(\omega|D, M_i) = P(D|\omega, M_i)P(\omega|M_i)/P(D|M_i)$ é pontuda no parâmetro mais provável $\hat{\omega}$, de forma que a integral pode ser aproximada para

$$P(D|M_i) \simeq P(D|\hat{\omega}, M_i)P(\hat{\omega}|M_i)\Delta\omega,$$

onde $P(D|\hat{\omega}, M_i)$ representa a evidência da MV e $P(\hat{\omega}|M_i)\Delta\omega$ o *fator de Ockham*.

Antes dos dados chegarem, o modelo M_i tem um intervalo $\Delta\omega_0$ para a escolha dos parâmetros ω (veja a figura 1.2). Depois, ao se considerar D_t, o intervalo dos possíveis ω vai para $\Delta\omega_t$. Como $\Delta\omega$ representa o volume do espaço dos parâmetros em acordo com D, apenas para facilitar a compreensão, podemos considerar

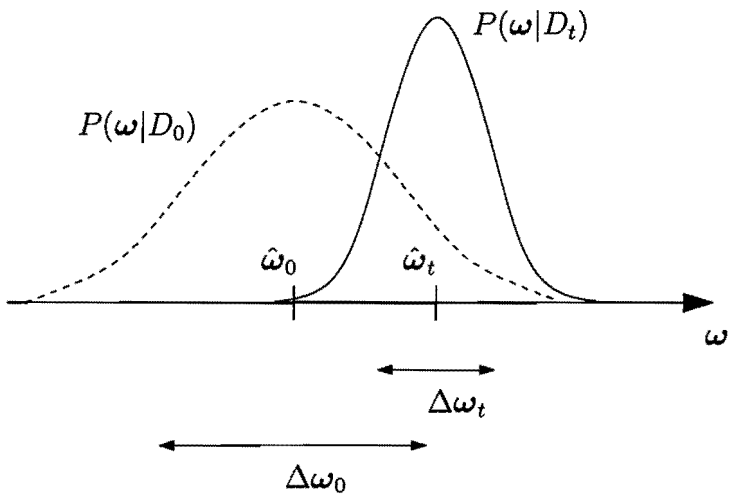

Figura 1.2: O fator de Ockham.

$P(\hat{\omega}|M_i) \simeq 1/\Delta\omega_0$, o que nos leva a

$$\text{Fator de Ockham} = \frac{\Delta\omega_t}{\Delta\omega_0}.$$

Pela última expressão, temos que os modelos mais complexos são penalizados com um fator de Ockham menor que o dos modelos mais simples. Por exemplo, modelos de ajuste de curvas, ou regressão, com muitos parâmetros livres têm um intervalo $\Delta\omega_0$ muito grande, de maneira que se ajustarão a qualquer curva com um desvio padrão muito baixo, ou seja $\Delta\omega_t$ muito pequeno. Assim, as evidências oriundas de modelagens bayesianas são naturalmente moduladas em favorecimento do modelo mais simples que satisfaça as informações desejadas.

Com isso, após esse breve ensaio sobre princípios de otimização e modelagem, vemos que a teoria bayesiana das probabilidades não é simplesmente uma alternativa para se reobter a teoria freqüencista, mas uma poderosa ferramenta que contém fundamentos matemáticos rigorosos e características, ou até mesmo podemos dizer natureza, propícias para a construção de máquinas processadoras de informações, ou seja, inteligência artificial.

1.2 Redes neurais artificiais.

Com o crescimento do conhecimento sobre a estrutura do cérebro, assim como sua relação funcional com o raciocínio humano, cresceu também o interesse sobre

questões envolvendo o processamento de informações.

No início do século XX d.C., conforme citamos anteriormente, a rede de neurônios do cérebro já era conhecida e tida como responsável pelo processamento de informações — embora não se conhecesse o seu funcionamento. Além disso, também era conhecido o comportamento tudo-ou-nada dos neurônios, ou seja, que eles disparavam quando a soma ponderada dos sinais recebidos de outros neurônios superava um certo limiar. Motivados por essas descobertas, em 1943 dois cientistas, Warren McCulloch e Walter Pitts, propuseram um modelo de redes de neurônios unificando os estudos da neurofisiologia e da lógica (matemática). Em seu trabalho "A logical calculus of the ideas immanent in nervous activity", eles tentaram entender como o cérebro poderia produzir padrões altamente complexos, utilizando uma rede de elementos de decisão binária que seriam uma versão simplificada dos neurônios biológicos.

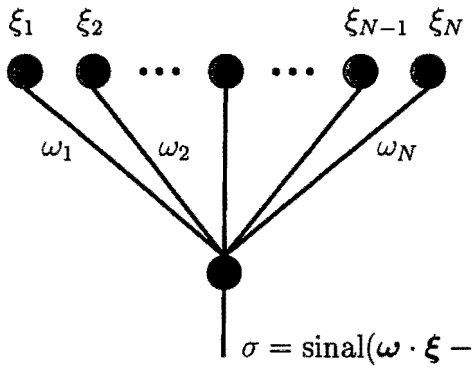

Figura 1.3: O neurônio de McCullooch e Pitts.

À semelhança do neurônio biológico, os neurônios de McCulloch e Pitts possuíam (ver figura 1.3): dendritos, por onde entravam as informações ξ^i's; sinapses ω^i's que ponderavam os ξ^i's; um limiar de disparo b; um corpo que processava a soma ponderada $\sum_i^N \omega^i \xi^i + b$ por meio de uma regra tudo-ou-nada[15] e um axônio que enviava o sinal de resposta σ para outros neurônios. Com a junção dessas redes eles concluíram que poderiam realizar quaisquer função computável e tarefas semelhante ao computador proposto por Alan Turing em 1936, bastando escolher a configuração das sinapses $\boldsymbol{\omega}$ adequadamente. Este foi um resultado muito significativo, de maneira que muitos cientistas têm esse trabalho como um marco do nascimento da disciplina

[15]A regra tudo-ou-nada pode ser obtida pela função $f(x) = 1/2 + \text{sinal}(x - b)/2$, gerando $\sigma \in \{0, 1\}$ que pode ser facilmente mapeado para $\sigma \in \{-1, 1\}$

de inteligência artificial.

A grande capacidade de processamento das redes de McCulloch e Pitts, que identificamos de uma maneira mais geral como redes neurais artificiais (RNA), dependia entretanto da escolha particular dos pesos sinápticos. Isto trouxe uma nova questão a ser estudada: *como escolher os pesos ω^i's de maneira que determinadas tarefas ou mapeamentos sejam realizados pela rede?* Uma das grandes respostas a essa pergunta foi apresentada por Frank Rosenblatt em 1958 [Rosenblatt 1958]. Naquele trabalho Rosenblatt introduziu um modelo de RNA com um algoritmo embutido para a obtenção dos pesos sinápticos, denominada por ele *perceptron.*

O algoritmo do perceptron utilizava os pares de informações $\{\sigma, \boldsymbol{\xi}\}$'s, tipo estímulo-reação (entrada-saída), e era da forma $\hat{\omega}^i_{\mu+1} = \hat{\omega}^i_\mu + \Delta$, onde $\hat{\boldsymbol{\omega}}$ representa a estimativa do "vetor sináptico" gerador dos pares $\{\sigma, \boldsymbol{\xi}\}$ e

$$\Delta = \begin{cases} \sigma_{\mu+1}\xi^i_{\mu+1} & \text{se} \quad \sigma_{\mu+1}\hat{\boldsymbol{\omega}} \cdot \boldsymbol{\xi}_{\mu+1} < 0 \\ 0 & \text{se} \quad \sigma_{\mu+1}\hat{\boldsymbol{\omega}} \cdot \boldsymbol{\xi}_{\mu+1} \geq 0. \end{cases} \tag{1.4}$$

Com a demonstração da convergência de $\hat{\boldsymbol{\omega}}$ para $\bar{\boldsymbol{\omega}}$, sendo $\bar{\boldsymbol{\omega}}$ o perceptron gerador do conjunto de informações $D = \{\sigma, \boldsymbol{\xi}\}$, Rosenblatt mostrou que tais redes podiam generalizar conceitos por meio de um processo de *aprendizagem.* Isto trouxe grande atenção para o estudo de RNA ampliando sua motivação, pois além de ser utilizado como um modelo para o estudo e compreensão dos princípios que o cérebro usa para trabalhar, também incentiva o estudo sobre a construção de máquinas que sejam capazes de realizar tarefas complexas e inferências a partir de informações indiretas e/ou incompletas.

Após aproximadamente sessenta anos, a questão levantada no trabalho de McCulloch e Pitts ainda continua sendo exaustivamente estudada. Os avanços obtidos nesse período são inegáveis, no entanto ainda há muito a se descobrir sobre os princípios fundamentais utilizados por nossa rede de neurônios no processamento de nossos raciocínios.

A seguir, introduzimos alguns conceitos e definições necessários para uma boa compreensão do trabalho aqui apresentado.

1.2.1 Aprendizagem numa rede neural artificial.

A propriedade que é de importância primordial para uma RNA é a sua habilidade de "aprender" a partir de seu ambiente (informações contidas em D) e melhorar o

seu desempenho através da "aprendizagem". No contexto de inteligência artificial essa aprendizagem na verdade é apenas um processo pelo qual os parâmetros livres da rede são adaptados através do processamento e/ou tratamento das informações fornecidas D, garantindo sua convergência para a resposta desejada.

O desempenho obtido em tal processo depende tanto da arquitetura da rede, ou seja, quais neurônios estão conectados a quais, quanto da distribuição dos valores de suas sinapses. De forma que, para fazer a RNA realizar alguma tarefa específica, é necessário escolher como os neurônios são conectados uns aos outros e distribuir os pesos sinápticos adequadamente[16]. O processo de aprendizagem em tais redes acontece então por meio de um processo iterativo sobre as informações de entrada, quando a rede busca uma configuração que solucione o problema proposto, generalizando-o. De acordo com as informações de entradas temos três tipos de aprendizagem: *Supervisionado, Não Supervisionado* e *com Reforço.* Na aprendizagem supervisionada nas informações de entrada constam as respostas desejadas às perguntas. Assim, temos o que chamamos de cenário *professor–aprendiz*, diferente da não supervisionada, onde não há um *supervisor* com as respostas desejadas. No caso de aprendizagem por reforço o supervisor apenas estima as respostas, pois não possui as respostas corretas.

A resposta desejada para rede neural é tratada como um vetor $\boldsymbol{\sigma}$, determinado por uma função de transferência $\mathcal{M}$ desconhecida que atua no vetor com os dados de entrada $\boldsymbol{\xi}$, geralmente de dimensão bem maior que $\boldsymbol{\sigma}$. Então

$$\boldsymbol{\sigma} = \mathcal{M}(\boldsymbol{\xi}),$$

onde tal mapeamento pode ser determinístico ou estocástico, de forma a possibilitar a corrupção de alguns dados por ruído.

O ato de aprender é então colocado como um problema de otimização por introduzir uma função erro, a qual mede a qualidade da aproximação da rede à função $\mathcal{M}$. Assim o processo de aprendizagem implica achar uma configuração das sinapses da rede que minimize a função erro.

As diferentes maneiras como as sinapses da rede são modificadas são definidas como algoritmo de aprendizagem.

O processo de interação da rede com as informações pode ser feito de muitas

[16] As conexões determinam como é possível um neurônio influenciar outro, e o pesos das sinapses dizem quão grande é tal influência.

formas, no entanto, dentre elas há dois limites interessantes: o *off-line* e o *on-line*. A classificação em *off-line* ocorre quando as informações de entrada são processadas em conjunto e a cada nova informação a rede revê todas as outras anteriores para só então atualizar o sistema, procurando a situação mais adequada para o conjunto inteiro de treinamento[17], o que pode ser facilmente associado à idéia de equilíbrio termodinâmico. Em oposição, temos o algoritmo que é classificado como *on-line*, quando a atualização é feita sempre que chega um novo dado, sem consultar o conjunto de treinamento como um todo, mas apenas o último dado e alguns parâmetros da rede, o que também pode ser facilmente associado a um processo fora de equilíbrio termodinâmico. Com tais associações torna-se possível a utilização das ferramentas da Mecânica Estatística no estudo de tais sistemas, visto também que o problema de aprendizagem pode ser visualizado como um processo de minimização de uma função erro (energia).

A necessidade de uma aprendizagem on-line surge imediatamente quando se precisa tomar decisões rápidas, não se tendo tempo para analisar todas as informações disponíveis. Tais situações são extremamente comuns em nossa vida diária, quando nos baseamos intensamente em nossos conhecimentos mais recentes. Por exemplo, podemos citar o caso de uma pessoa sozinha na selva vendo entre as folhas de uma moita um tecido amarelo com manchas pretas. No primeiro momento a pessoa não tem a plena certeza se aquilo que está na moita é apenas um pedaço de pano ou se a pele de uma onça, mas se ela ficar ali até ter essa certeza, e se realmente for uma onça, essa informação não lhe será mais tão útil uma vez que não conseguirá escapar do ataque da onça, por outro lado, se a pessoa correr imediatamente após ver o tecido entre as folhas, poderá escapar. Assim vemos que em determinadas situações é preferível termos processamentos rápidos com conclusões razoáveis do que termos conclusões exatas mas com longo tempo de processamento. É claro que em tais processamentos mais rápidos (processos on-line) perdemos uma certa quantia de informação que não foi utilizada, mas surpreendentemente mostra-se [Engel et al 2001] que algoritmos baseados em aprendizagem on-line podem encontrar um desempenho semelhante aos baseados em off-line, quando o número de neurônios é consideravelmente grande ($N \to \infty$). Além disso, mesmo que possamos construir máquinas onde os tempos de processamento para algoritmos off-line sejam toleráveis, ainda temos uma grande motivação para o estudo de algoritmos on-line: a aprendizagem

[17] Que consta de todas as informações já utilizadas mais a que chega a cada instante.

de conceitos não-estacionários.

A aprendizagem de conceitos não-estacionários ocorre quando o mapeamento $\mathcal{M}$ muda durante a coleta dos dados do treinamento, de maneira contínua ou momentânea. Isso significa que utilizar informações sem levar em consideração suas respectivas *idades* levará a grandes erros na inferência, tanto maior quanto menor for o tempo característico dessas mudanças ou a quantidade de informações velhas utilizadas para predizer o novo conceito.

Num processo de aprendizagem tipicamente off-line, todos os dados são tratados em paralelo, de forma a não haver distinção, *a priori*, de suas respectivas idades. Num algoritmo on-line, por natureza os dados são processados individualmente na seqüência de sua coleta, o que torna propício a inclusão de módulos que observem a idade dos dados.

2

Redes neurais bayesianas.

No capítulo anterior vimos que a abordagem bayesiana naturalmente satisfaz os principais princípios de modelagem, como o princípio de Ockham e a regularização de Tikhonov. Além disso, também vimos que a idéia de probabilidade como sendo um grau de crença não só abrange a interpretação freqüencista e pode ser apresentada com o mesmo rigor matemático [Bernado et al 2000], como também está inserida na motivação inicial da construção de uma teoria para o raciocínio humano. No entanto sabemos que os méritos da teoria bayesiana perante outras abordagens não serão estabelecidos puramente por discussões filosóficas, mas apenas por demonstrações efetivas de sua superioridade em contextos práticos, ou seja, na construção de modelos e algoritmos de aprendizagem.

Naturalmente não existe um algoritmo que seja sempre melhor para qualquer tipo de problema, ou melhor, independente do problema. Porém, dentro de uma mesma subclasse de problemas é de fundamental importância saber se existe uma maneira ótima de se utilizar as informações contidas em D (o conjunto de Dados). A existência de tal nos leva também à existência de um limite superior no desempenho que não pode ser ultrapassado por nenhum algoritmo.

Esses desempenhos são quantificados por meio do *erro de treinamento*, que é definido como sendo a fração de exemplos mal classificados de D, e do *erro de generalização* que nos dá a probabilidade de um exemplo qualquer (contido ou não em D) ser mal classificado. Se definirmos o conjunto de parâmetros que regulam o mapeamento $\mathcal{M}$ como $\bar{\omega} \in \Re^N$ (o vetor sináptico de uma RNA), podemos dizer

que o erro de generalização quantifica a distância entre $\bar{\omega}$ e um outro conjunto de parâmetros $\hat{\omega} \in \Re^N$. Além disso, podemos também imaginar um espaço Ω que contenha todos os possíveis candidatos para $\bar{\omega}$, ou melhor, *versões* delimitadas por D. Dessa forma o processo de aprendizagem consiste na busca em Ω pelo melhor candidato sobre o critério de minimização do erro de generalização.

Para o problema de classificação binária (CB) por superfícies lineares usando redes neurais artificiais esse limite inferior no erro de generalização existe, ou seja, é conhecido, e é alcançado pelo algoritmo bayesiano off-line.

Um problema CB consiste em: dado um conjunto de pontos $D_p = \{\boldsymbol{\xi}_\mu, \sigma_\mu\}_{\mu=1}^p$, sendo $\boldsymbol{\xi}_\mu \in \Re^N$ e $\sigma_\mu \in \{1; -1\}$, encontrar uma superfície que separe os pontos $\boldsymbol{\xi}_\mu$ com $\sigma_\mu = 1$ dos $\boldsymbol{\xi}_\nu$ com $\sigma_\nu = -1$ (veja a figura 2.1). Salientamos entretanto que apesar da teoria apresentada ser geral o suficiente para ser aplicada em problemas CB com qualquer tipo de superfície, nos restringimos apenas ao caso de superfícies lineares[1], ou seja, hiperplanos.

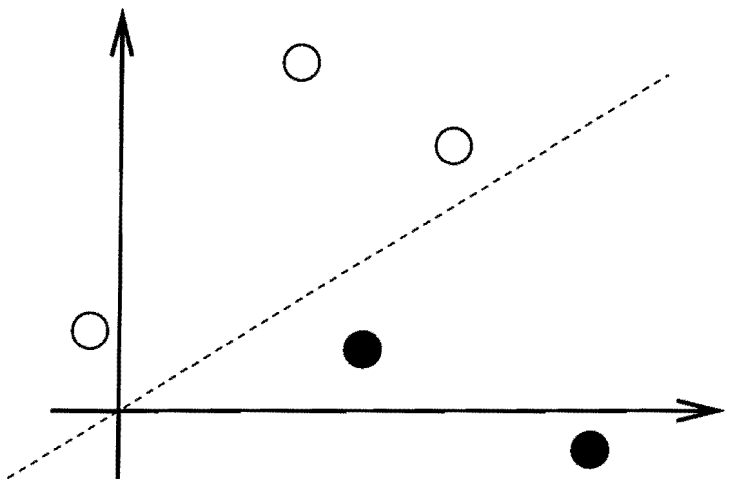

Figura 2.1: Problema de classificação binária. Os pontos acima da linha tracejada (bolas brancas) recebem a classificação $\sigma = 1$ e os que ficam abaixo (bolas pretas) recebem $\sigma = -1$.

[1]O caso de superfícies não-lineares é tratado de maneira equivalente por meio de transformações no espaço de entrada (nos vetores $\boldsymbol{\xi}_\mu$), podendo-se trabalhar diretamente com os vetores transformados ou indiretamente por meio do núcleo da transformação. Sistemas que utilizam essa estratégia são conhecidos por *máquinas núcleo* (kernel machines) ou *máquinas de vetores de suporte* (support vectors machines).

No processo de aprendizagem cada elemento de D delimita a região de localização de $\bar{\omega}$ em Ω. Após μ exemplos, a predição de $\sigma_{\mu+1}$ no algoritmo bayesiano off-line (ABOff) é dada pela resposta majoritária dos elementos de Ω condizentes com os μ primeiros exemplos. Isso equivale a construir uma distribuição posterior $P(\omega|D)$ e ter como uma estimativa para o professor $\hat{\omega} = \int \mathrm{d}\omega P(\omega|D)\omega$, ou seja, a estimativa para o professor será equivalente ao *centro de massa* do espaço de versões delimitado por D [Engel et al 2001].

O bom desempenho do ABOff é até esperado, visto que sua predição satisfaz os vínculos impostos por D e minimiza a chance de erro por meio do voto majoritário. Entretanto seu desempenho não é apenas bom, mas considerado como o próprio limite inferior do erro de generalização e_g para o problema CB, com um comportamento assintótico dado por $e_g = 0,44/\alpha$ [Opper et al 1990, Engel et al 2001], sendo α a razão entre o número μ de exemplos utilizados e a dimensão N do espaço de entrada ($\alpha = \mu/N$).

Embora estejamos nos limitando ao problema de CB, no decorrer da última década foram apresentados vários trabalhos[2] mostrando modelos bayesianos com melhores desempenhos que algoritmos oriundos de outras abordagens (como a maximização da verossimilhança, por exemplo) em vários outros tipos de problemas de aprendizagem ou inferência estatística (regressão, reconstrução de imagens, reconhecimento de voz, ...).

Na verdade, o crescimento da taxa de aparição desses trabalhos deve-se principalmente ao crescimento da capacidade de armazenamento e da velocidade de processamento dos computadores, bem como ao fácil acesso a tais. Assim, a dificuldade de se calcular integrais $N-$dimensionais (o centro de massa $\hat{\omega}$, por exemplo) é vencida por meio de métodos aproximativos baseados em simulação numérica [Andrieu et al 2003]. Contudo, o uso desses métodos não deve ser visto como um substituto das soluções analíticas mesmo quando não for possível encontrar uma solução analítica exata, até porque tais métodos são aprimorados por meio de tratamentos e aproximações analíticos. Além do mais, em situações não-estacionárias o tratamento off-line dos dados não é adequado.

Como em inferência estatística todo o conhecimento sobre um dado sistema é representado por distribuições de probabilidades, a estratégia natural para se obter um

[2]Veja os anais das conferências NIPS (Neural Information Processing Systems – - http://www.nips.cc/) e [Amaral et al 2004].

tratamento bayesiano aproximativo, porém analítico, é construir uma distribuição posterior por meio de um tratamento on-line. Dessa forma, em princípio, mantêm-se tanto a versatilidade dos algoritmos on-line quanto a boa qualidade bayesiana no tratamento das informações.

2.1 O algoritmo on-line.

De acordo com o teorema de Bayes [Bernado et al 2000], a distribuição de probabilidades para os candidatos ω's posterior a D_μ é dada por[3]

$$P(\omega|D_\mu) = \frac{P(\omega)P(D_\mu|\omega)}{\int \mathrm{d}^N\omega' P(\omega')P(D_\mu|\omega')}, \tag{2.1}$$

sendo $P(\omega)$ a distribuição *a priori* de ω, $P(D_\mu|\omega)$ a verossimilhança dos $\mu-$elementos do conjunto D_μ para um dado ω e $\int \mathrm{d}^N\omega' P(\omega')P(D_\mu|\omega')$ a marginal $P(D_\mu)$.

Na chegada de nova informação $y_{\mu+1} = (\xi_{\mu+1}, \sigma_{\mu+1})$ temos

$$P(\omega|y_{\mu+1}, D_\mu) = \frac{P(\omega|D_\mu)P(y_{\mu+1}|\omega, D_\mu)}{\int \mathrm{d}^N\omega' P(\omega'|D_\mu)P(y_{\mu+1}|\omega', D_\mu)},$$

que para o caso $P(y_{\mu+1}|D_\mu) = P(y_{\mu+1})$ [4] resulta em

$$P(\omega|D_{\mu+1}) = \frac{P(\omega|D_\mu)P(y_{\mu+1}|\omega)}{\int \mathrm{d}^N\omega' P(\omega'|D_\mu)P(y_{\mu+1}|\omega')}. \tag{2.2}$$

Perceba que, embora a equação 2.2 nos diga como atualizar a distribuição posterior frente a chegada de $y_{\mu+1}$, a predição de $\sigma_{\mu+2}$ continua sendo feita por meio da integral $\int \mathrm{d}^N\omega P(\omega|D_{\mu+1})\omega$ que trata em paralelo todos os y_μ's, ou seja, off-line. Então, como adaptá-la à um processo onde apenas o novo dado $y_{\mu+1}$ e "pouca coisa" a mais seja suficiente, sem a necessidade de se reutilizar os dados velhos $y_{\nu<\mu+1}$?

Uma boa resposta a tal pergunta é simplesmente projetar a distribuição dos pesos sinápticos ω num espaço de funções tratáveis [Opper 1998]. A idéia é substituir médias sobre distribuições complicadas por médias sobre distribuições aproximadas mais simples. No entanto, para o procedimento ser significativo é necessário que a distribuição aproximada capture da distribuição verdadeira os fatores essenciais para a aprendizagem.

[3] $P(A,B) = P(A|B)P(B) = P(B|A)P(A) \Rightarrow P(A|B) = P(A)P(B|A)/P(B) = P(A)P(B|A)/\int P(A')P(B|A')\mathrm{d}A'$.

[4] Para os elementos (ξ_μ, σ_μ)'s independente e identicamente distribuídos.

Em geral, a atualização da distribuição posterior feita por meio de 2.2 a retira do espaço de funções inicial $\mathcal{G}$, onde é possível tratar de maneira razoável as integrais envolvidas. Assim, após cada passo de atualização é necessário projetar $P(\omega|D_{\mu+1}) = S_{\mu+1}$ em $\mathcal{G}$ (ver figura 2.2). Para minimizar a perda de informação ao projetarmos $S_{\mu+1}$ de S em $\mathcal{G}$, escolhemos o elemento $\mathcal{G}_{\mu+1}$ que minimiza a divergência de Kullback-Leibler:

$$D_{KL}(S_{\mu+1}, \mathcal{G}_{\mu+1}) = \left\langle \ln \frac{S_{\mu+1}(\omega|D_{\mu+1})}{\mathcal{G}_{\mu+1}(\omega)} \right\rangle_{\omega \sim S_{\mu+1}}. \tag{2.3}$$

Assim, o algoritmo bayesiano on-line (ABOn) é divido em duas etapas:

Atualizar Frente a chegada de $y_{\mu+1}$, $\mathcal{G}_\mu(\omega)$ é atualizada por meio da regra de Bayes

$$S_{\mu+1}(\omega) = \frac{\mathcal{G}_\mu(\omega) P(y_{\mu+1}|\omega)}{\int \mathrm{d}^N\omega' \mathcal{G}_\mu(\omega') P(y_{\mu+1}|\omega')}. \tag{2.4}$$

Projetar Seleciona-se $\mathcal{G}_{\mu+1} \in \mathcal{G}$ tal que $\delta_{\mathcal{G}_{\mu+1}} D_{KL}(S_{\mu+1}, \mathcal{G}_{\mu+1}) = 0$.

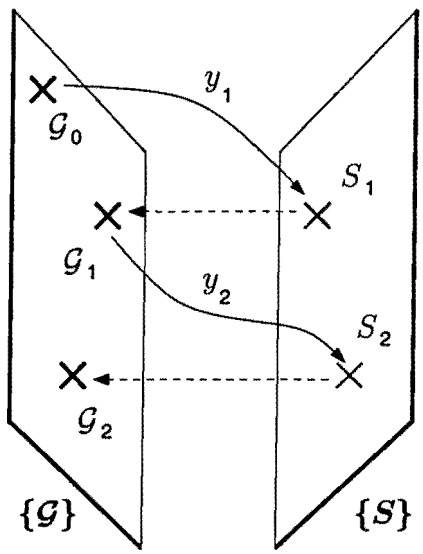

Figura 2.2: Atualização da distribuição posterior por meio da minimização da divergência de Kullback-Leibler. A cada informação $y_{\mu+1}$, $\mathcal{G}_\mu$ é atualizada por meio de 2.4 indo para $S_{\mu+1}$. Se $S_{\mu+1} \notin \mathcal{G}$, $S_{\mu+1}$ é substituída por $\mathcal{G}_{\mu+1}$ que minimiza a divergência de Kullback-Leibler.

Como toda modelagem bayesiana, para aplicarmos o ABOn num dado problema é necessário definirmos inicialmente a distribuição *a priori* $P(\boldsymbol{\omega})$ e a verossimilhança $P(y_\mu|\boldsymbol{\omega})$. No ABOn, a escolha da família de funções $\mathcal{G}$ define automaticamente $P(\boldsymbol{\omega}) = \mathcal{G}_0(\boldsymbol{\omega})$, ou seja, as informações *a priori* sobre o vetor sináptico $\boldsymbol{\omega}$ devem ser incluídas na escolha de $\mathcal{G}$. Enquanto que informações sobre a estrutura da rede e sobre os dados, são geralmente postos na verossimilhança $P(y_\mu|\boldsymbol{\omega})$.

Escolhendo $\mathcal{G}$ como a família das gaussianas, temos que a variação de $D_{KL}(S_\mu, \mathcal{G}_\mu)$ é feita com relação a média e a covariância da gaussiana.

$$\begin{cases} \mathcal{G}_\mu(\boldsymbol{\omega}) &= \exp\left\{-(\boldsymbol{\omega}-\hat{\boldsymbol{\omega}}_\mu)\cdot C_\mu^{-1}\cdot(\boldsymbol{\omega}-\hat{\boldsymbol{\omega}}_\mu)/2\right\}/\sqrt{(2\pi)^N|C_\mu|} \\ \delta_{\mathcal{G}_\mu} &= \sum_i^N \delta_{\hat{\omega}_\mu^i} + \sum_{i,j}^N \delta_{C_\mu^{ij}} \end{cases}$$

Por sua vez, a minimização da divergência de Kullback-Leibler é obtida por meio da equivalência entre os dois primeiros cumulantes[5] de S_μ e $\mathcal{G}_\mu$.

$$\hat{\omega}_\mu^i = \left\langle \omega^i \right\rangle_{\omega\sim S_\mu} \tag{2.5}$$

$$C_\mu^{ij} = \left\langle (\omega^i - \hat{\omega}_\mu^i)(\omega^j - \hat{\omega}_\mu^j) \right\rangle_{\omega\sim S_\mu} \tag{2.6}$$

Assim, uma vez que a estimativa dos pesos sinápticos é dada pelo centro de massa da distribuição posterior, no algoritmo ABOn com a família de funções gaussianas esta é o próprio vetor $\hat{\boldsymbol{\omega}}_\mu$. Além disso, a sua equação de atualização é obtida por meio das equações 2.5, 2.6 e 2.4.

Aplicando a equação 2.4 na 2.5, encontramos que

$$\begin{aligned} \hat{\omega}_{\mu+1}^i &= \frac{\int \mathrm{d}^N\omega\ \omega^i \exp\left\{-(\boldsymbol{\omega}-\hat{\boldsymbol{\omega}}_\mu)\cdot C_\mu^{-1}\cdot(\boldsymbol{\omega}-\hat{\boldsymbol{\omega}}_\mu)/2\right\} P(y_{\mu+1}|\boldsymbol{\omega})}{\int \mathrm{d}^N\omega \exp\left\{-(\boldsymbol{\omega}-\hat{\boldsymbol{\omega}}_\mu)\cdot C_\mu^{-1}\cdot(\boldsymbol{\omega}-\hat{\boldsymbol{\omega}}_\mu)/2\right\} P(y_{\mu+1}|\boldsymbol{\omega})} \\ &= \hat{\omega}_\mu^i + \frac{\int \mathrm{d}^N u\ u^i \exp\left\{-\boldsymbol{u}\cdot C_\mu^{-1}\cdot\boldsymbol{u}/2\right\} P(y_{\mu+1}|\boldsymbol{u}+\hat{\boldsymbol{\omega}}_\mu)}{\int \mathrm{d}^N u \exp\left\{-\boldsymbol{u}\cdot C_\mu^{-1}\cdot\boldsymbol{u}/2\right\} P(y_{\mu+1}|\boldsymbol{u}+\hat{\boldsymbol{\omega}}_\mu)}. \end{aligned} \tag{2.7}$$

Onde realizamos a mudança de variável $\boldsymbol{u} = \boldsymbol{\omega} - \hat{\boldsymbol{\omega}}_\mu$ e que por meio da identidade

[5]Seja $g(v)$ a função característica de uma dada distribuição $P(x)$, ou seja, $g(v) = \int \exp\{-ivx\}P(x)\mathrm{d}x$, os cumulantes k_n de $P(x)$ são definidos pela expansão $\ln g(v) = \sum_{n=1}^{\infty}(iv)^n k_n/n!$. Assim, temos que o primeiro cumulante, k_1, é igual ao valor médio de x e o segundo, k_2, a sua variância.

$$\begin{cases} u^i = \sum_l u^l \delta_{i,l} = \sum_{l,j} C_\mu^{ij} [C_\mu^{-1}]^{jl}\, u^l \\ \sum_l [C_\mu^{-1}]^{il}\, u^l \exp\{-\frac{1}{2}\boldsymbol{u}\cdot C_\mu^{-1}\cdot\boldsymbol{u}\} = -\partial_{u^i} \exp\{-\frac{1}{2}\boldsymbol{u}\cdot C_\mu^{-1}\cdot\boldsymbol{u}\}, \end{cases}$$

ou seja,

$$u^i \exp\{-\frac{1}{2}\boldsymbol{u}\cdot C_\mu^{-1}\cdot\boldsymbol{u}\} = -\sum_l C_\mu^{il}\partial_{u^l} \exp\{-\frac{1}{2}\boldsymbol{u}\cdot C_\mu^{-1}\cdot\boldsymbol{u}\},$$

transforma-se em

$$\hat{\omega}_{\mu+1}^i = \hat{\omega}_\mu^i + \sum_l C_\mu^{il} \frac{\int e^{-\frac{1}{2}\boldsymbol{u}\cdot C_\mu^{-1}\cdot\boldsymbol{u}} \partial_{u^l} P(y_{\mu+1}|\boldsymbol{u}+\hat{\boldsymbol{\omega}}_\mu)\mathrm{d}^N u}{\int e^{-\frac{1}{2}\boldsymbol{u}\cdot C_\mu^{-1}\cdot\boldsymbol{u}} P(y_{\mu+1}|\boldsymbol{u}+\hat{\boldsymbol{\omega}}_\mu)\mathrm{d}^N u}.$$

Observando que $\partial_{u^l} f(\boldsymbol{u}+\hat{\boldsymbol{\omega}}_\mu) = \partial_{\hat{\omega}_\mu^l} f(\boldsymbol{u}+\hat{\boldsymbol{\omega}}_\mu)$, podemos reescrever a última expressão como

$$\hat{\boldsymbol{\omega}}_{\mu+1} = \hat{\boldsymbol{\omega}}_\mu + C_\mu \cdot \nabla_{\hat{\boldsymbol{\omega}}_\mu} \ln \langle P(y_{\mu+1}|\boldsymbol{u}+\hat{\boldsymbol{\omega}}_\mu)\rangle_{\boldsymbol{u}}. \tag{2.8}$$

Para a equação 2.6 temos

$$C_{\mu+1}^{ij} = \int (u^i - \Delta\hat{\omega}_{\mu+1}^i)(u^j - \Delta\hat{\omega}_{\mu+1}^j) S_{\mu+1}(\boldsymbol{u}+\hat{\boldsymbol{\omega}}_\mu)\mathrm{d}^N u,$$

sendo $\Delta\hat{\omega}_{\mu+1}^i = \hat{\omega}_{\mu+1}^i - \hat{\omega}_\mu^i$ e que ao utilizarmos 2.4 e a identidade

$$u^i u^j e^{-\frac{1}{2}\boldsymbol{u}\cdot C_\mu^{-1}\cdot\boldsymbol{u}} = C_\mu^{ij} e^{-\frac{1}{2}\boldsymbol{u}\cdot C_\mu^{-1}\cdot\boldsymbol{u}} + \sum_{k,l} C_\mu^{ik} C_\mu^{jl} \partial_{u^k}\partial_{u^l} e^{-\frac{1}{2}\boldsymbol{u}\cdot C_\mu^{-1}\cdot\boldsymbol{u}},$$

torna-se

$$C_{\mu+1} = C_\mu + C_\mu \cdot \nabla_{\hat{\boldsymbol{\omega}}_\mu} \otimes \nabla_{\hat{\boldsymbol{\omega}}_\mu} \cdot C_\mu \ln \langle P(y_{t+1}|\boldsymbol{u}+\hat{\boldsymbol{\omega}}_\mu)\rangle_{\boldsymbol{u}}. \tag{2.9}$$

Uma vez obtido $\hat{\boldsymbol{\omega}}_{\mu+1}$ por meio de 2.8, a estimativa para $\sigma_{\mu+2}$ é obtida pelo sinal do produto $\hat{\boldsymbol{\omega}}_{\mu+1}\cdot\boldsymbol{\xi}_{\mu+2}$, ou seja, $\sigma_{\mu+2} = \mathrm{sinal}(\hat{\boldsymbol{\omega}}_{\mu+1}\cdot\boldsymbol{\xi}_{\mu+2})$, e da mesma forma todos as seguintes entradas $\boldsymbol{\xi}_\nu$ com $\nu \geqslant \mu+1$. Assim, as equações 2.8 e 2.9 determinam o nosso ABOn para o problema CB, que denominaremos por *ABOn tensorial.*

Como podemos perceber, o ABOn tensorial se assemelha aos populares algoritmos de descida pelo gradiente [Haykin 2003, Duda et al 2001]. Nesses algoritmos define-se uma função custo $\mathcal{E}(\boldsymbol{\omega})$ com os pares y_μ como parâmetros. Essa função, denominada *potencial de erro*, quantifica uma distância entre a estimativa $\hat{\sigma}_\mu$ a resposta conhecida σ_μ dados $\boldsymbol{\omega}$ e $\boldsymbol{\xi}_\mu$. Assim o vetor sináptico ótimo será aquele que satisfizer a condição $\mathcal{E}(\bar{\boldsymbol{\omega}}) \leqslant \mathcal{E}(\boldsymbol{\omega})$ para todo $\boldsymbol{\omega} \in \Omega$, ou seja um ponto de mínimo.

Algoritmos tipo descida pelo gradiente (DG) são escritos na forma[6]

[6]Definimos $\mathcal{E}_{\mu+1} \equiv \mathcal{E}(\hat{\boldsymbol{\omega}}_\mu; \sigma_{\mu+1}, \boldsymbol{\xi}_{\mu+1})$.

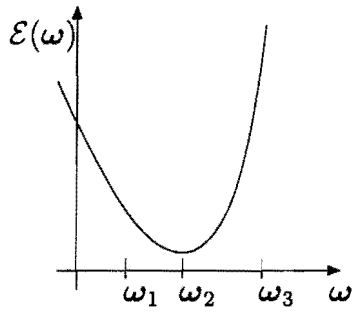

Figura 2.3: Potencial de erro $\mathcal{E}(\omega)$.

$$\hat{\omega}_{\mu+1} = \hat{\omega}_\mu - \eta \nabla_{\hat{\omega}_\mu} \mathcal{E}_{\mu+1}, \tag{2.10}$$

onde $\eta \geqslant 0$ é denominado *parâmetro de taxa de aprendizagem*, ou simplesmente *taxa de aprendizagem*.

Pela figura 2.3 podemos entender o funcionamento de um algoritmo tipo 2.10.

Quando a estimativa de vetor sináptico estiver a esquerda de ω_2 (por exemplo $\hat{\omega}_\mu = \omega_1$) teremos que o seu gradiente será negativo, visto que ω_2 é um ponto de mínimo. Assim, $-\eta \nabla_{\hat{\omega}_\mu} \mathcal{E}_{\mu+1} > 0$ e $\hat{\omega}_{\mu+1}$ ficará mais próximo de ω_2. Para $\hat{\omega}_\mu = \omega_3$ (a esquerda de ω_2), $-\eta \nabla_{\hat{\omega}_\mu} \mathcal{E}_{\mu+1} < 0$ e novamente $\hat{\omega}_{\mu+1}$ ficará mais próximo de ω_2. Entretanto, é necessário ressaltarmos a influência de η na convergência de $\hat{\omega}_\mu$ para o ponto de mínimo $\bar{\omega}$. Se η for muito pequeno, a convergência de $\hat{\omega}_\mu$ será desnecessariamente lenta. Por outro lado, se η for muito grande, $\hat{\omega}_\mu$ oscilará em torno da solução com uma grande possibilidade de não-convergência (tanto maior, quanto maior for η). Além disso, seria bom se $\eta \equiv \eta_\mu$, ou seja, que η variasse durante o treinamento.

O ponto de partida de um processo de aprendizagem em redes neurais é geralmente escolhido aleatoriamente, de forma que pode-se escolher um $\hat{\omega}_0$ muito longe do ponto de mínimo. Assim, torna-se interessante[7] que η_μ seja grande para que o algoritmo vá mais rápido à região de mínimo; por outro lado, passada a fase inicial é interessante que η_μ seja pequeno para que o algoritmo não caia em oscilação permanente.

Para contornar esse problema da dinâmica de η_μ, uma das propostas mais populares é encontrada por meio da expansão de $\mathcal{E}(\omega)$ até segunda ordem em torno de

[7]Com relação a velocidade de convergência.

$\hat{\omega}_\mu$:

$$\mathcal{E}(\omega) \simeq \mathcal{E}(\hat{\omega}_\mu) + (\omega - \hat{\omega}_\mu) \cdot \nabla\mathcal{E} + \frac{1}{2}(\omega - \hat{\omega}_\mu) \cdot H \cdot (\omega - \hat{\omega}_\mu). \tag{2.11}$$

Sendo $\nabla\mathcal{E}$ o gradiente com relação a ω, calculado em $\hat{\omega}_\mu$ e H a matriz *Hessiana* cujos elementos são dados por $H_{ij} \equiv \partial^2\mathcal{E}/\partial\omega^i\partial\omega^j$, também calculados em $\hat{\omega}_\mu$. Minimizando 2.11 com relação a ω ($\hat{\omega}_{\mu+1}$ será o vetor ω que miminize $\mathcal{E}(\omega)$), encontramos a solução[8]

$$\hat{\omega}_{\mu+1} = \hat{\omega}_\mu - H_\mu^{-1}\nabla_{\hat{\omega}}\mathcal{E}_{\mu+1}. \tag{2.12}$$

O algoritmo definido pela equação 2.12 é conhecido por *algoritmo de Newton* e apresenta em geral um desempenho melhor que os algoritmos DG simples, cujo η_μ é escalar. Entretanto, nem sempre é possível garantir que H_μ seja não-singular durante todo o treinamento, o que inutiliza o algoritmo. Além disso, temos a necessidade de inverter H_μ, ou seja, um custo adicional de $\mathcal{O}(N^3)$. Em contrapartida, o ABOn tensorial por construção possui C_μ positivo definido[9] e é obtido diretamente da equação 2.9, sem o custo de $\mathcal{O}(N^3)$.

2.1.1 Desempenho assintótico.

Façamos agora um rápido estudo sobre o comportamento do algoritmo descrito pelas equações 2.8 e 2.9 para $\mu \gg 1$.

Como a correção $\Delta\hat{\omega}_{\mu+1}$ é regulada por C_μ, para sabermos o grau de eficiência e precisão do algoritmo em questão é necessário sabermos anteriormente a forma assintótica de C_μ.

Definindo $M^{kl} = \partial_{\hat{\omega}_\mu^k}\partial_{\hat{\omega}_\mu^l} \ln \langle P(y_{\mu+1}|\boldsymbol{u} + \hat{\omega}_\mu)\rangle_{\boldsymbol{u}}$ e assumindo que para grandes 'μ' as mudanças temporais de C_μ são pequenas, podemos introduzir uma evolução temporal contínua para $C(\mu)$ partindo de 2.9:

$$\frac{\mathrm{d}C}{\mathrm{d}\mu} = C \cdot M \cdot C, \tag{2.13}$$

[8]Como $\nabla\mathcal{E}$ é o gradiente calculado em $\hat{\omega}_\mu$, ele também é equivalente a $\nabla_{\hat{\omega}}\mathcal{E}_{\mu+1}$ e, de forma semelhante, $H \equiv H_\mu$.

[9]$\boldsymbol{v} \cdot C \cdot \boldsymbol{v} \geqslant 0$ para todo $\boldsymbol{v} \in \Re^N$.

que pode ser reescrita como[10]

$$\frac{\mathrm{d}C^{-1}}{\mathrm{d}\mu} = -M,$$

ou seja,

$$C^{-1}(\mu) - C^{-1}(0) = -\int_0^\mu M(\mu')\mathrm{d}\mu'. \tag{2.14}$$

Para incluirmos em nossa análise o caso em que o modelo é mal especificado, suporemos que $Q(y_\mu|\bar{\omega})$ não pertença à mesma família de $P(y_\mu|\omega)$, ou melhor, que os elementos y_μ sejam gerados por um distribuição qualquer $Q(y_\mu)$.

Devido a ausência de métodos gerais para o cálculo explícito de uma convergência global para quaisquer algoritmos on-line, iremos nos limitar à situação na qual a RNA está próxima a um ponto fixo atrativo $\bar{\omega}$, o que nos possibilitará obter uma expressão exata para a convergência assintótica do algoritmo.

O ponto fixo $\bar{\omega}$ corresponde a um mínimo local da divergência entre $Q(y)$ e $P(y|\omega)$ dada por[11]

$$D_{KL}(Q, P_\omega) = \int Q(y) \ln\left(\frac{Q(y)}{P(y|\omega)}\right) \mathrm{d}y,$$

ou seja, satisfaz a igualdade:

$$\int Q(y)\left[\partial_{\omega^i} \ln P(y|\omega)\right]_{\bar{\omega}} \mathrm{d}y = 0. \tag{2.15}$$

Levando em consideração que para μ grande a distribuição posterior $\mathcal{G}(\omega|D_\mu)$ estará fortemente concentrada em torno de seu máximo em $\bar{\omega}$, podemos negligenciar a sua largura de forma a termos $\langle P(y|\omega)\rangle_\omega$ equivalente à $P(y|\bar{\omega})$. Assim, dividindo a equação 2.14 por μ e tomando o limite $\mu \to \infty$, encontraremos:

$$\begin{aligned}\lim_{\mu\to\infty} \frac{C^{-1}_{ij}(\mu)}{\mu} &= \lim_{\mu\to\infty} -\frac{1}{\mu}\int_0^\mu \left[\partial_{\omega^i}\partial_{\omega^j} \ln P(y(\mu)|\omega)\right]_{\bar{\omega}} \mathrm{d}\mu \\ &= -\int Q(y)\left[\partial_{\omega^i}\partial_{\omega^j} \ln P(y|\omega)\right]_{\bar{\omega}} \mathrm{d}y,\end{aligned} \tag{2.16}$$

considerando que o sistema seja ergódico.

A equação 2.16 nos mostra que para tempos suficientemente longos as flutuações de $\hat{\omega}_\mu$ em torno do ponto fixo $\bar{\omega}$ podem ser negligenciadas, uma vez que $C \simeq$

[10] $\mathrm{d}C^{-1}/\mathrm{d}\mu = -C^{-1} \cdot \mathrm{d}C/\mathrm{d}\mu \cdot C^{-1}$.

[11] Sempre que conveniente e equivalente, trocaremos a soma $\sum_x$ por $\int \mathrm{d}x P(x)$.

$[\int(\cdot\cdot)]^{-1}/\mu$. Isso significa que uma vez dentro da bacia de atração, ou seja, num vale que contenha $\bar{\omega}$, o termo de correção $\Delta\hat{\omega}_{\mu+1}$ será cada vez menor, decaindo a zero e garantindo assim que $\hat{\omega}_\mu$ não escape da região de solução. Além disso, quando $Q(y) = P(y|\bar{\omega})$ teremos

$$-\int P(y|\bar{\omega})\left[\partial_{\omega^i}\partial_{\omega^j}\ln P(y|\omega)\right]_{\bar{\omega}}\mathrm{d}y \;=\; \int \frac{1}{P(y|\bar{\omega})}\left[\partial_{\omega_i}P(y|\omega)\partial_{\omega_j}P(y|\omega)\right]_{\bar{\omega}}\mathrm{d}y,$$

onde utilizamos 2.15 e que pode ser reescrita como[12]

$$\lim_{\mu\to\infty}\frac{C_{ij}^{-1}(\mu)}{\mu} = \int dyP(y|\bar{\omega})\partial_i\ln P(y|\bar{\omega})\partial_j\ln P(y|\bar{\omega}). \tag{2.17}$$

A integral contida na última expressão é conhecida como *matriz de informação de Fisher*, $\mathcal{J}^{ij} = \int dyP(y|\omega)\partial_{\omega^i}\ln P(y|\omega)\partial_{\omega^j}\ln P(y|\omega)$. Para algoritmos que apresentam $\langle\hat{\omega}_\mu\rangle_{D_\mu} = \bar{\omega}$ (para estimadores sem viés), a velocidade de aproximação de $\bar{\omega}$ por $\hat{\omega}_\mu$ é limitada proporcionalmente à inversa de $\mathcal{J}$. Essa desigualdade é conhecida por *Desigualdade de Rao-Cramér*:

$$\langle(\hat{\omega}_\mu - \bar{\omega})\otimes(\hat{\omega}_\mu - \bar{\omega})\rangle_{D_\mu} \geqslant \mathcal{J}^{-1}/\mu. \tag{2.18}$$

Uma vez que a velocidade de aproximação de $\hat{\omega}_\mu$ a $\bar{\omega}$ pelo ABOff é limitada por 2.18 [Opper 1998], é interessante sabermos também como tal grandeza se comporta para o algoritmo ABOn tensorial. Para tanto, definimos o erro $e_\mu = \hat{\omega}_\mu - \bar{\omega}$ e assumimos novamente que $\hat{\omega}_\mu$ está próximo a $\bar{\omega}$, com uma distribuição estreita.

Expandindo $\partial_{\hat{\omega}^i_\mu}\ln P(y_{\mu+1}|\hat{\omega}_\mu)$ em torno de $\bar{\omega}$ e considerando apenas os termos até primeira ordem em e_μ, temos:

$$\partial_{\hat{\omega}^i_\mu}\ln P(y_{\mu+1}|\hat{\omega}_\mu) = \partial_{\hat{\omega}^i_\mu}\ln P(y_{\mu+1}|\hat{\omega}_\mu)\Big|_{\bar{\omega}} + \sum_k \partial_{\hat{\omega}^k_\mu}\partial_{\hat{\omega}^i_\mu}\ln P(y_{\mu+1}|\hat{\omega}_\mu)\Big|_{\bar{\omega}} e^k_\mu + ...,$$

o que nos leva a

$$\Delta\hat{\omega}^i_{\mu+1} = \Delta e^i_{\mu+1} = \sum_l C^{il}_\mu\partial_l\ln P(y_{\mu+1}|\bar{\omega}) + \sum_{lk} C^{il}_\mu e^k\partial_k\partial_l\ln P(y_{\mu+1}|\bar{\omega}). \tag{2.19}$$

Agora é só calcularmos a média da equação 2.19 com relação ao conjunto $D_{\mu+1}$ e, por meio do mesmo argumento utilizado anteriormente, passarmos de tempos discretos para contínuos.

[12]Para simplificar a notação trocamos expressões do tipo $[\partial_{\omega^i}F(\omega)]_{\omega=\bar{\omega}}$ por $\partial_i F(\bar{\omega})$.

Definindo $\hat{e}^i = \langle e^i \rangle_{D_{\mu+1}}$, teremos que

$$\begin{aligned} \frac{\mathrm{d}\hat{e}^i}{\mathrm{d}\mu} &= \sum_l C^{il} \langle \partial_l \ln P(y|\bar{\omega}) \rangle_y + \sum_{lk} C^{il} \hat{e}^k \langle \partial_k \partial_l \ln P(y|\bar{\omega}) \rangle_y \\ &= \sum_l \frac{A^{-1}_{il}}{\mu} \langle \partial_l \ln P(y|\bar{\omega}) \rangle_y - \frac{\hat{e}^i}{\mu}, \end{aligned} \tag{2.20}$$

válido no limite $\mu \to \infty$ e onde definimos a matriz

$$A^{il} = -\int Q(y) \partial_i \partial_l \ln P(y|\bar{\omega}) dy. \tag{2.21}$$

Devido a 2.15 temos que o primeiro termo do lado direito da equação 2.20 desaparece, de forma que o erro linear médio $\hat{e}^i$ decai com $1/\mu$, ou seja, o valor médio $\langle \hat{\omega}^i \rangle_D$ tende com o passar do tempo ao ponto fixo $\bar{\omega}^i$ — que satisfaz 2.15.

Por fim, vejamos como se comporta o erro quadrático médio,

$$E^{ij} = \langle (\hat{\omega}_i - \bar{\omega}_i)(\hat{\omega}_j - \bar{\omega}_j) \rangle_D .$$

Escrevendo $\boldsymbol{e}_{\mu+1} = \boldsymbol{e}_\mu + \Delta \boldsymbol{e}_{\mu+1}$, temos que

$$\boldsymbol{e}_{\mu+1} \otimes \boldsymbol{e}_{\mu+1} - \boldsymbol{e}_\mu \otimes \boldsymbol{e}_\mu = \Delta \boldsymbol{e}_{\mu+1} \otimes \Delta \boldsymbol{e}_{\mu+1} + \Delta \boldsymbol{e}_{\mu+1} \otimes \boldsymbol{e}_\mu + \boldsymbol{e}_\mu \otimes \Delta \boldsymbol{e}_{\mu+1},$$

ou seja,

$$\begin{aligned} \frac{\mathrm{d}E}{\mathrm{d}\mu} &= C \cdot B \cdot C - C \cdot A \cdot E - E \cdot A \cdot C \\ &= \frac{1}{\mu^2} A^{-1} \cdot B \cdot A^{-1} - \frac{2}{\mu} E, \end{aligned} \tag{2.22}$$

onde definimos

$$B = \int Q(y) \partial_i \ln P(y|\bar{\omega}) \partial_l \ln P(y|\bar{\omega}) dy \tag{2.23}$$

e negligenciamos termos proporcionais a $\hat{e}^i C^{jk} \langle \partial_k \ln P(y|\bar{\omega}) \rangle_y$.

A equação 2.22 é resolvida por

$$E = \frac{1}{\mu} A^{-1} \cdot B \cdot A^{-1}, \tag{2.24}$$

para $\mu \to \infty$, ou seja, para um mínimo local $\bar{\omega}$ de $-\int Q(y) P(y|\omega) \mathrm{d}y$ sempre teremos o decaimento ótimo $\propto 1/\mu$ para o erro quadrático médio. Além disso, quando o modelo for bem especificado teremos $B = A = \mathcal{J}$, ou seja,

$$E \overset{\mu\to\infty}{\longrightarrow} \frac{1}{\mu}\mathcal{J}^{-1}(\bar{\omega}).$$

Isso significa que o ABOn tensorial possui a mesma eficiência assintótica que o ABOff — segundo o critério de Rao-Cramér.

Embora esses resultados sejam interessantes, nem sempre é possível inferir a habilidade de predição de novos dados partindo deles. Por isso, no que segue, sempre faremos nossas análises com base no erro de generalização, uma vez que nossos dados serão gerados artificialmente, ou seja, conhecemos $\bar{\omega}$.

2.2 Utilizando uma distribuição gibbsiana.

A identificação do potencial $\mathcal{E}_{\mu+1}$ em 2.8 nos fornece uma nova ferramenta para estudarmos as relações entre algoritmos off-line e os seus respectivos algoritmos on-line [Caticha et al 2001].

Em geral, os algoritmos de descida pelo gradiente usam os mesmos potenciais dos algoritmos off-line. Entretanto foi observado que nem sempre um potencial bem sucedido[13] em uma aprendizagem off-line será o melhor numa versão on-line [Kinouchi et al 1992].

No ABOn tensorial, a conexão entre o algoritmo off-line e o on-line é feita por meio da verossimilhança, ou seja, numa abordagem off-line a informação dos dados é obtida através de $P(y|\omega)$, enquanto que na abordagem on-line é através de $\mathcal{E} = \ln \langle P(y|\boldsymbol{u}+\omega)\rangle_{\boldsymbol{u}}$. Assim, o algoritmo bayesiano procura uma solução que maximiza $P(y|\omega)$, enquanto que num algoritmo de descida pelo gradiente, a solução é dada pela minimização do potencial de erro $E(\omega; D) = \sum_{y\in D} V(\omega; y)$.

A equivalência entre a maximização da verossimilhança e a minimização do potencial de erro é obtida por meio da escolha [Levin et al 1990]:

$$P(y|\omega) = \frac{\exp\{-\beta V(\omega; y)\}}{\int \mathrm{d}^N\omega \exp\{-\beta V(\omega; y)\}}. \tag{2.25}$$

[13] Com um bom desempenho ou até mesmo com um *desempenho ótimo*. Dizemos que um dado algoritmo possui desempenho ótimo quando o seu erro de generalização for assintoticamente semelhante ao do ABOff. Para o problema CB, isso equivale a termos $e_g(\alpha) \overset{\alpha\to\infty}{\longrightarrow} c/\alpha$, sendo c constante.

Com a introdução da distribuição gibbsiana 2.25, reformulamos o nosso problema de aprendizagem como um típico problema da mecânica estatística. O parâmetro β funciona como o inverso de uma temperatura T. No limite $T \to 0$, temos que $P(y|\boldsymbol{\omega})$ será não-nulo somente quando $V(\boldsymbol{\omega}; y) = 0$, ou seja, quando não houver erros. Isso significa uma confiança plena nos dados fornecidos, bem como na existência de uma solução $\bar{\boldsymbol{\omega}}$ cujo erro de treinamento $e_t = \sum_{y \in D} V(\omega; y)$ seja nulo.

Aplicando a equação 2.25 em 2.4 encontramos a distribuição

$$\mathcal{S}_{\mu+1}(\boldsymbol{\omega}) = \frac{\mathcal{G}_\mu(\boldsymbol{\omega}) e^{-\beta V(\boldsymbol{\omega}; y_{\mu+1})}}{\int \mathrm{d}^N \omega' \mathcal{G}_\mu(\boldsymbol{\omega}') e^{-\beta V(\boldsymbol{\omega}'; y_{\mu+1})}}, \tag{2.26}$$

a partir da qual seguimos o mesmo procedimento representado na figura 2.2, impondo a igualdade entre os dois primeiros cumulantes de $\mathcal{S}_{\mu+1}(\boldsymbol{\omega})$ e $\mathcal{G}_\mu(\boldsymbol{\omega})$. Assim, o algoritmo para uma distribuição gibbsiana é:

$$\hat{\boldsymbol{\omega}}_{\mu+1} = \hat{\boldsymbol{\omega}}_\mu + C_\mu \cdot \nabla_{\hat{\boldsymbol{\omega}}_\mu} \ln \left\langle e^{-\beta V(\boldsymbol{u} + \hat{\boldsymbol{\omega}}_\mu; y_{\mu+1})} \right\rangle_{\boldsymbol{u}}; \tag{2.27}$$

$$C_{\mu+1} = C_\mu + C_\mu \cdot \nabla_{\hat{\boldsymbol{\omega}}_\mu} \otimes \nabla_{\hat{\boldsymbol{\omega}}_\mu} \cdot C_\mu \ln \left\langle e^{-\beta V(\boldsymbol{u} + \hat{\boldsymbol{\omega}}_\mu; y_{\mu+1})} \right\rangle_{\boldsymbol{u}}. \tag{2.28}$$

Além do ajuste tensorial no gradiente, a grande diferença entre o algoritmo definido pelas equações 2.27 e 2.28 e os algoritmos DG tradicionais é a média da exponencial do potencial. Perceba que se não tivéssemos a média em $\boldsymbol{u}$ teríamos $\Delta\hat{\boldsymbol{\omega}}_{\mu+1} = -\beta C_\mu \cdot \nabla_{\hat{\boldsymbol{\omega}}_\mu} V(\hat{\boldsymbol{\omega}}_\mu; y_{\mu+1})$, ou seja, o potencial on-line seria equivalente ao off-line e a taxa de aprendizagem seria regulada por β.

Para entendermos melhor a mudança do potencial pelo algoritmo, consideremos potenciais com a dependência: $V(\hat{\boldsymbol{\omega}}_\mu; \boldsymbol{\xi}_{\mu+1}, \sigma_{\mu+1}) = V(\sigma_{\mu+1} \boldsymbol{\omega} \cdot \boldsymbol{\xi}_{\mu+1})$. Apenas para facilitar as contas, introduzamos duas novas varáveis: $\tau_\mu = \sigma_{\mu+1} \hat{\boldsymbol{\omega}}_\mu \cdot \boldsymbol{\xi}_{\mu+1}/\sqrt{N}$ e $\lambda = \sigma_{\mu+1} \boldsymbol{\omega} \cdot \boldsymbol{\xi}_{\mu+1}/\sqrt{N} = \tau_\mu + \sigma_{\mu+1} \boldsymbol{u} \cdot \boldsymbol{\xi}_{\mu+1}/\sqrt{N}$. Com isso temos que

$$\begin{aligned}
\left\langle e^{-\beta V(\boldsymbol{u} + \hat{\boldsymbol{\omega}}_\mu; y_{\mu+1})} \right\rangle_{\boldsymbol{u}} &= \int \frac{e^{-\frac{1}{2} \boldsymbol{u} \cdot C_\mu^{-1} \cdot \boldsymbol{u}}}{\sqrt{(2\pi)^N |C_\mu|}} e^{-\beta V(\lambda)} \delta(\lambda - \tau_\mu - \sigma_{\mu+1} \boldsymbol{u} \cdot \boldsymbol{\xi}_{\mu+1}/\sqrt{N}) \mathrm{d}\lambda \mathrm{d}^N u \\
&= \int \frac{e^{-\frac{1}{2} \boldsymbol{u} \cdot C_\mu^{-1} \cdot \boldsymbol{u} - \beta V(\lambda)}}{2\pi \sqrt{(2\pi)^N |C_\mu|}} e^{i\theta(\lambda - \tau_\mu - \sigma_{\mu+1} \boldsymbol{u} \cdot \boldsymbol{\xi}_{\mu+1}/\sqrt{N})} \mathrm{d}\theta \mathrm{d}\lambda \mathrm{d}^N u \\
&= \int \exp\{-\theta^2 \boldsymbol{\xi}_{\mu+1} \cdot C_\mu \cdot \boldsymbol{\xi}_{\mu+1}/2N + i\theta(\lambda - \tau_\mu) - \beta V(\lambda)\} \frac{\mathrm{d}\theta \mathrm{d}\lambda}{2\pi} \\
&= \frac{1}{\sqrt{2\pi \mathsf{x}_\mu}} \int \exp\left(-\beta V(\lambda) - \frac{(\lambda - \tau_\mu)^2}{2\mathsf{x}_\mu}\right) \mathrm{d}\lambda,
\end{aligned} \tag{2.29}$$

sendo $\mathsf{x}_\mu = \boldsymbol{\xi}_{\mu+1} \cdot C_\mu \cdot \boldsymbol{\xi}_{\mu+1}/N$.

No limite de temperatura zero, podemos definir um novo $\mathsf{x}'_\mu = \beta \mathsf{x}_\mu$ de forma a encontrarmos pelo método do ponto de sela

$$\left\langle e^{-\beta V(\boldsymbol{u}+\hat{\boldsymbol{\omega}}_\mu; y_{\mu+1})} \right\rangle_{\boldsymbol{u}} \simeq \frac{1}{\sqrt{2\pi \mathsf{x}_\mu}} \exp\left(-\beta V(\lambda_0) - \beta \frac{(\lambda_0 - \tau_\mu)^2}{2\mathsf{x}'_\mu} \right), \tag{2.30}$$

com λ_0 satisfazendo:

$$\left[\frac{\partial V(\lambda)}{\partial \lambda} + \frac{\lambda - \tau_\mu}{\mathsf{x}'_\mu} \right]_{\lambda=\lambda_0} = 0.$$

Portanto, podemos escrever o potencial on-line induzido pelo algoritmo ABOn como

$$\mathcal{E}_{\mu+1} \equiv V(\lambda_0) + \frac{(\lambda_0 - \tau_\mu)^2}{2\mathsf{x}'_\mu}. \tag{2.31}$$

Observando que $\partial_{\hat{\omega}^i_\mu} = \frac{1}{N}\sigma_{\mu+1}\xi^i_{\mu+1}\partial_{\tau_\mu}$ e que $[\partial_\lambda V]_{\lambda_0} = [\partial_\tau \mathcal{E}]_{\tau_\mu}$ finalmente chegamos ao algoritmo[14]

$$\hat{\boldsymbol{\omega}}_{\mu+1} = \hat{\boldsymbol{\omega}}_\mu - \frac{1}{N}\sigma_{\mu+1}\tilde{C}_\mu \cdot \boldsymbol{\xi}_{\mu+1}\partial_{\tau_\mu}\mathcal{E}_{\mu+1}, \tag{2.32}$$

$$\tilde{C}_{\mu+1} = \tilde{C}_\mu - \frac{1}{N}\tilde{C}_\mu \cdot \boldsymbol{\xi}_{\mu+1} \otimes \boldsymbol{\xi}_{\mu+1}\tilde{C}_\mu \partial^2_{\tau_\mu}\mathcal{E}_{\mu+1}. \tag{2.33}$$

Consideremos agora que o potencial induzido seja plano para grandes magnitudes da estabilidade τ_μ. Para valores negativos de τ_μ, $\mathcal{E}_{\mu+1}$ satura num valor positivo enquanto que para valores positivos de τ_μ ele vai a zero, decaindo monotonicamene na região de transição de forma que sua segunda derivada seja positiva se a nova informação for classificada corretamente e negativa se não for. Isso significa que o algoritmo muda seu comportamento de acordo com uma estimativa de seu próprio desempenho. Se comete um erro (classifica incorretamente a informação), ele reage aumentando sua estimativa da taxa de aprendizagem. Se classifica corretamente o novo exemplo, ele reduz a taxa de aprendizagem, reduzindo por conseqüência as próximas correções dos pesos $\hat{\omega}^i$'s.

2.2.1 O potencial de Rosenblatt.

Vejamos agora como se comporta o nosso algoritmo 2.27, 2.28 com a utilização do potencial de Rosenblatt.

[14] $\tilde{C}_\mu \equiv \beta C_\mu$

O potencial de Rosenblatt é definido por $V(\lambda) = -\lambda\Theta(-\lambda)$, sendo λ a estabilidade $\sigma\boldsymbol{\omega}\cdot\boldsymbol{\xi}/\sqrt{N}$. A motivação de tal escolha se deve ao fato de termos um erro de generalização assintótico $e_g \propto \alpha^{-1}$ para algoritmos off-line e $e_g \propto \alpha^{-1/3}$ para algoritmos on-line que usam diretamente o mesmo potencial [Engel et al 2001], ou seja, uma nítida diferença no desempenho de algoritmos baseados num mesmo potencial, mas com diferentes esquemas de iteração dos dados (off-line e on-line).

Para obtermos o potencial on-line induzido devemos primeiro calcular a integral

$$\left\langle e^{-\beta V(\boldsymbol{u}+\hat{\boldsymbol{\omega}}_\mu;y_{\mu+1})}\right\rangle_{\boldsymbol{u}} = \frac{1}{\sqrt{2\pi \mathrm{x}_\mu}}\int_{-\infty}^{\infty}\exp\left(\beta\lambda\Theta(-\lambda) - \frac{(\lambda-\tau_\mu)^2}{2\mathrm{x}_\mu}\right)\mathrm{d}\lambda, \tag{2.34}$$

que no limite $\beta\to\infty$ torna-se

$$\left\langle e^{-\beta V(\boldsymbol{u}+\hat{\boldsymbol{\omega}}_\mu;y_{\mu+1})}\right\rangle_{\boldsymbol{u}} = \frac{1}{2}\mathrm{erfc}\left(-\tau_\mu/\sqrt{2\mathrm{x}_\mu}\right). \tag{2.35}$$

Portanto, a menos de uma constante, o nosso potencial on-line para $V(\lambda) = -\lambda\Theta(-\lambda)$ é $\mathcal{E}_{\mu+1} = -\ln\mathrm{erfc}\left(-\tau_\mu/\sqrt{2\mathrm{x}_\mu}\right)$, que por sua vez resulta no algoritmo:

$$\hat{\boldsymbol{\omega}}_{\mu+1} = \hat{\boldsymbol{\omega}}_\mu + \sigma_{\mu+1}\sqrt{\frac{2}{N\pi\mathrm{x}_\mu}}C_\mu\cdot\boldsymbol{\xi}_{\mu+1}\mathcal{F}\left(\frac{-\tau_\mu}{\sqrt{2\mathrm{x}_\mu}}\right) \tag{2.36}$$

$$C_{\mu+1} = C_\mu - \frac{1}{N\mathrm{x}_\mu}C_\mu\cdot\boldsymbol{\xi}_{\mu+1}\otimes\boldsymbol{\xi}_{\mu+1}\cdot C_\mu\mathcal{F}\left(\frac{-\tau_\mu}{\sqrt{2\mathrm{x}_\mu}}\right)\left[\tau\sqrt{\frac{2}{\pi\mathrm{x}_\mu}} + \frac{2}{\pi}\mathcal{F}\left(\frac{-\tau_\mu}{\sqrt{2\mathrm{x}_\mu}}\right)\right], \tag{2.37}$$

sendo $\mathcal{F}(x) = e^{-x^2}/\mathrm{erfc}(x)$.

Vejamos a dinâmica das equações 2.36, 2.37 por meio das derivadas de $\mathcal{E}(x) = -\ln\mathrm{erfc}\,(-x)$ desenhadas na figura 2.4.

Ao contrário do potencial off-line, que faz correções apenas na presença de erros, vemos pela curva $-\mathrm{d}_x\mathcal{E}(x)$ que $\Delta\hat{\boldsymbol{\omega}}_{\mu+1}$ difere de zero tanto para estabilidades negativas como para positivas. Entretanto, quando a magnitude da estabilidade é grande ($x > 3$) não há correções a serem feitas. Isso significa que um exemplo cuja classificação é óbvia (quando o ponto está muito distante da superfície de classificação) não traz informação relevante e, portanto, não acarreta nenhuma mudança. Da mesma forma, a magnitude de $\Delta C_{\mu+1}$ vai a zero para estabilidades grandes e varia de sinal de acordo com a orientação dos vetores de entrada — devido ao termo $C_\mu\cdot\boldsymbol{\xi}_{\mu+1}\otimes\boldsymbol{\xi}_{\mu+1}\cdot C_\mu$.

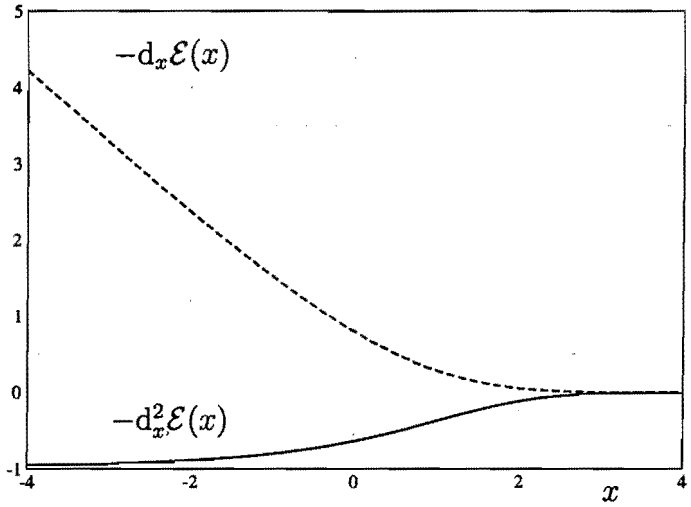

Figura 2.4: Primeira (linha tracejada) e segunda (linha cheia) derivadas do potencial on-line induzido de Rosenblatt para $\beta \to \infty$.

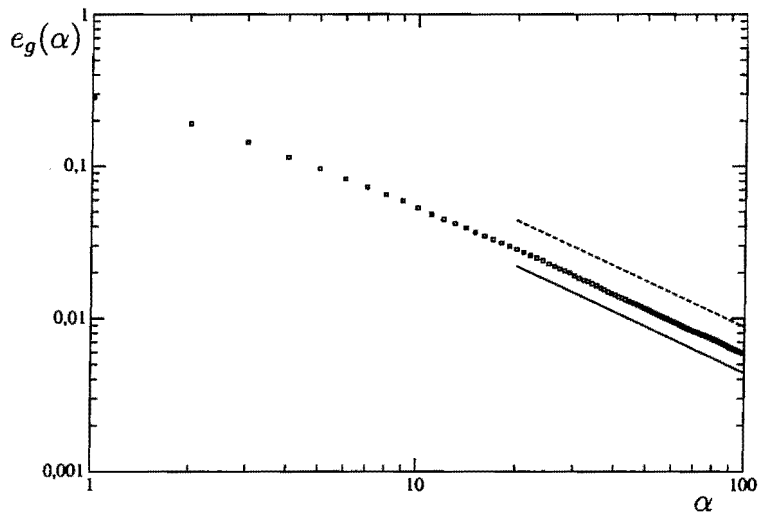

Figura 2.5: Erro de generalização para o perceptron com regra constante. Os pontos quadrados foram obtidos pelo algoritmo ABOn tensorial. A linha tracejada (superior) representa a curva $0,88/\alpha$ e a linha cheia (inferior) $0,44/\alpha$.

Na figura 2.5 apresentamos o erro de generalização obtido pelo algoritmo 2.36[15],

[15]Todas as curvas experimentais apresentadas nessa tese foram obtidas pela média de 100 simu-

2.37. A curva obtida pelo algoritmo foi ajustada assintoticamente por $e_g \simeq 0,54/\alpha^{0,98}$ com desvio padrão de $\mathcal{O}(10^{-2})$ em ambos parâmetros (coeficiente e expoente). Isso nos mostra que o algoritmo não só é eficiente, de acordo com o critério de Rao-Cramér, como também apresenta um desempenho ótimo. Além disso, contrariamente a conclusão de [Solla et al 1998], observamos que o coeficiente do erro de generalização não converge para 0,88 e sim para algum valor abaixo dele.

O erro assintótico $e_g = 0,88/\alpha$ é alcançado pelo o algoritmo ótimo (AO), obtido por meio do método variacional (ver apêndice B), que apresenta uma forma muito próxima de 2.36:

$$\hat{\omega}_{\mu+1} = \hat{\omega}_\mu + \sigma_{\mu+1}\boldsymbol{\xi}_{\mu+1}||\hat{\omega}_\mu||\sqrt{\frac{2r_\mu}{\pi N^2}}\mathcal{F}\left(\frac{-t}{\sqrt{2r_\mu}}\right), \tag{2.38}$$

sendo $t_\mu = \tau_\mu/||\hat{\omega}_\mu||$, $r_\mu = (1-\rho_\mu^2)/\rho_\mu^2$ e $\rho_\mu = \hat{\omega}_\mu \cdot \bar{\omega}/||\hat{\omega}_\mu||\ ||\bar{\omega}||$.

Como podemos perceber, além do ajuste tensorial em $\boldsymbol{\xi}_{\mu+1}$, a diferença entre 2.38 e 2.36 é a troca de r_μ por x_μ. Na verdade, essa troca é muito conveniente para fins práticos, já que a grandeza r_μ geralmente não é acessível enquanto que x_μ é calculado pelo próprio algoritmo. Uma vez que ambos algoritmos compartilham do mesmo potencial on-line, é razoável o considerarmos como responsável pelo limite assintótico $e_g \propto \alpha^{-1}$, o que nos leva a pergunta: *qual seria a grande contribuição do ajuste tensorial ?*

2.2.2 Simplificando o algoritmo: gaussianas esféricas.

Para respondermos à última questão escolhamos $\mathcal{G}$ como sendo a família de gaussianas esféricas, ou seja, $\mathcal{G}_\mu(\omega) = \exp\{-||(\omega - \hat{\omega}_\mu)||^2/2\zeta_\mu\}/\sqrt{2\pi\zeta_\mu}$, com $\hat{\omega}_\mu \in \Re^N$ e $\zeta_\mu \in \Re_+$. O algoritmo, *ABOn escalar*, resultante de tal escolha é[16]

$$\hat{\omega}_{\mu+1} = \hat{\omega}_\mu + \sigma_{\mu+1}\boldsymbol{\xi}_{\mu+1}\sqrt{\frac{2\zeta_\mu}{\pi N}}\mathcal{F}\left(\frac{-\tau_\mu}{\sqrt{2\zeta_\mu}}\right); \tag{2.39}$$

$$\zeta_{\mu+1} = \zeta_\mu - \frac{1}{N}\zeta_\mu\left[\tau\sqrt{\frac{2}{\pi\zeta_\mu}} + \frac{2}{\pi}\mathcal{F}\left(\frac{-\tau_\mu}{\sqrt{2\zeta_\mu}}\right)\right]\mathcal{F}\left(\frac{-\tau_\mu}{\sqrt{2\zeta_\mu}}\right). \tag{2.40}$$

O processo de aprendizagem descrito pelo último par de equações[17] é um processo

lações com $N = 100$.

[16]Como estamos interessados no regime assintótico em α com $N \to \infty$, consideramos $\boldsymbol{\xi}\cdot\boldsymbol{\xi}/N = 1$ ($\xi_i \sim \mathcal{N}(0,1)$) na obtenção das equações 2.39 e 2.40.

[17]Na verdade, de uma maneira mais geral por equações do tipo 2.10

estocástico, visto que em cada passo ele recebe um vetor $\boldsymbol{\xi}_\mu$ tirado aleatoriamente de uma dada distribuição $P(\boldsymbol{\xi})$. Portanto, o procedimento tradicional para resolvermos sua dinâmica seria calcular a evolução da distribuição de probabilidades das variáveis de interesse. Entretanto, como estamos interessados no comportamento de tais variáveis no limite termodinâmico $N \to \infty$, é possível escrevermos diretamente as equações diferenciais de interesse e em seguida obtermos o comportamento assintótico do algoritmo.

As variáveis de interesse, nesse caso, são a norma do vetor sináptico $\hat{\boldsymbol{\omega}}_\mu$ e a sua correlação com o vetor professor $\bar{\boldsymbol{\omega}}$, uma vez que estamos utilizando o erro de generalização $e_g(\alpha) = \frac{1}{\pi}\text{acos}\ \rho(\alpha)$ como medida de desempenho (ver apêndice A).

Utilizando as normas reescaladas $Q_\mu = \hat{\boldsymbol{\omega}}_\mu \cdot \hat{\boldsymbol{\omega}}_\mu / N$ e $M = \bar{\boldsymbol{\omega}} \cdot \bar{\boldsymbol{\omega}}/N$, temos que

$$\rho_\mu = \frac{\hat{\boldsymbol{\omega}}_\mu \cdot \bar{\boldsymbol{\omega}}}{N\sqrt{QM}}.$$

Como parâmetro de tempo escolhemos $\alpha = \mu/N$ de forma a termos $\Delta\alpha = 1/N$, o que torna-se altamente conveniente para grandes valores de N e, portanto, para o limite contínuo desejado. Assim, partindo da equação 2.39 e definindo $\hat{\tau}_\mu = \tau_\mu/\sqrt{Q_\mu}$ encontramos

$$\begin{aligned} Q(\alpha + 1/n) - Q(\alpha) &= \frac{2}{N}\left[\sqrt{\frac{2Q(\alpha)}{\pi\zeta(\alpha)^{-1}}}\sigma(\alpha)\hat{\tau}(\alpha) + \frac{\zeta(\alpha)}{\pi}\mathcal{F}\left(-\hat{\tau}(\alpha)\sqrt{\frac{Q(\alpha)}{2\zeta(\alpha)}}\right)\right] \\ &\quad \mathcal{F}\left(-\hat{\tau}(\alpha)\sqrt{\frac{Q(\alpha)}{2\zeta(\alpha)}}\right), \end{aligned} \tag{2.41}$$

que nos leva a

$$\begin{aligned} \frac{Q(\alpha + \nu/N) - Q(\alpha)}{\nu/N} &= \frac{2}{\nu}\sum_{n=0}^{\nu-1}\left[\sqrt{\frac{2Q(\alpha+\frac{n}{N})}{\pi\zeta(\alpha+\frac{n}{N})^{-1}}}\sigma(\alpha+\frac{n}{N})\hat{\tau}(\alpha+\frac{n}{N})\right. \\ &\quad \left. + \frac{\zeta(\alpha+\frac{n}{N})}{\pi}\mathcal{F}\left(\frac{-\hat{\tau}(\alpha+\frac{n}{N})}{\sqrt{2\zeta(\alpha+\frac{n}{N})}}\right)\right]\mathcal{F}\left(\frac{-\hat{\tau}(\alpha+\frac{n}{N})}{\sqrt{2\zeta(\alpha+\frac{n}{N})}}\right). \end{aligned} \tag{2.42}$$

No limite $\nu \to \infty$ e $N \to \infty$ temos que: n/N torna-se contínuo; o lado esquerdo de 2.42 torna-se uma derivada temporal (com relação a α) e $\frac{1}{\nu}\sum_n^{\nu-1}$ torna-se uma integral sobre D, ou, de forma equivalente, sobre $\{\sigma, \hat{\tau}, \bar{\tau}\}$ (ver apêndice A). Com isso, chegamos a[18]

[18] Veja [Reents et al 1998] para uma demonstração mais formal.

$$\frac{\mathrm{d}Q}{\mathrm{d}\alpha} = 2\left\langle \left[\sqrt{\frac{2Q\zeta}{\pi}}\sigma\hat{\tau} + \frac{\zeta}{\pi}\mathcal{F}\left(\frac{-\hat{\tau}}{\sqrt{2\zeta}}\right)\right]\mathcal{F}\left(\frac{-\hat{\tau}}{\sqrt{2\zeta}}\right)\right\rangle_{\sigma,\hat{\tau}}, \tag{2.43}$$

Realizando a soma em σ, finalmente encontramos a equação diferencial para $Q(\alpha)$:

$$\frac{\mathrm{d}Q}{\mathrm{d}\alpha} = \frac{2\zeta}{\pi}\int_{-\infty}^{\infty}\mathrm{d}\hat{\tau}\frac{\mathrm{erfc}(-\hat{\tau}/\sqrt{r})}{\mathrm{erfc}(-\hat{\tau}\sqrt{Q/\zeta})}\left[2\hat{\tau}\sqrt{\frac{Q}{\zeta}} + \frac{e^{-Q\hat{\tau}^2/\zeta}}{\sqrt{\pi}\mathrm{erfc}(-\hat{\tau}\sqrt{Q/\zeta})}\right]e^{-(1+Q/\zeta)\hat{\tau}^2} \tag{2.44}$$

Seguindo o mesmo procedimento encontramos a equação diferencial para $\zeta(\alpha)$:

$$\frac{\mathrm{d}\zeta}{\mathrm{d}\alpha} = -\frac{2\zeta}{\pi}\int_{-\infty}^{\infty}\mathrm{d}\hat{\tau}\frac{\mathrm{erfc}(-\hat{\tau}/\sqrt{r})}{\mathrm{erfc}(-\hat{\tau}\sqrt{Q/\zeta})}\left[\hat{\tau}\sqrt{\frac{Q}{\zeta}} + \frac{e^{-Q\hat{\tau}^2/\zeta}}{\sqrt{\pi}\mathrm{erfc}(-\hat{\tau}\sqrt{Q/\zeta})}\right]e^{-(1+Q/\zeta)\hat{\tau}^2} \tag{2.45}$$

Por fim, nos resta obter a equação diferencial para ρ. Para tanto multiplicamos 2.39 por $\bar{\omega}$,

$$\rho' = (\bar{\omega}\cdot\hat{\omega} + \sqrt{\frac{2\zeta}{\pi}}\sigma'\bar{\tau}'\mathcal{F})/N\sqrt{Q'}, \tag{2.46}$$

estando as variáveis com $'$ em $(\alpha + 1/N)$ e as demais em α, com exceção de $\bar{\omega}$ que é constante durante o treinamento. Também introduzimos $\bar{\tau}(\alpha) = \bar{\omega}\cdot\xi(\alpha)/\sqrt{N}$ e $\mathcal{F} \equiv \mathcal{F}(-\hat{\tau}\sqrt{Q/2\zeta})$.

Por meio de 2.41 obtemos

$$Q'^{-1/2} \simeq Q^{-1/2}\left[1 - \frac{1}{NQ}\left(\sigma\hat{\tau}\sqrt{2Q\zeta/\pi} + \frac{\zeta}{\pi}\mathcal{F}\right)\mathcal{F}\right],$$

que, colocado em 2.46, resulta em

$$\frac{\mathrm{d}\rho}{\mathrm{d}\alpha} = \left\langle\sqrt{\frac{2\zeta}{\pi Q}}(\bar{\tau} - \rho\hat{\tau})\sigma\mathcal{F}\left(\frac{-\hat{\tau}}{\sqrt{2\zeta}}\right) - \frac{\rho\zeta}{\pi Q}\mathcal{F}^2\left(\frac{-\hat{\tau}}{\sqrt{2\zeta}}\right)\right\rangle_{\sigma,\bar{\tau},\hat{\tau}}$$

e que, por sua vez, ao integrarmos em $\bar{\tau}$ e somarmos em σ, torna-se

$$\begin{aligned}\frac{\mathrm{d}\rho}{\mathrm{d}\alpha} = \frac{2\rho}{\pi^{3/2}}\int_{-\infty}^{\infty}\mathrm{d}\hat{\tau}\frac{e^{-(1+Q/\zeta)\hat{\tau}^2}}{\mathrm{erfc}^2(-\hat{\tau}\sqrt{Q/\zeta})}\Bigg[&\sqrt{\frac{r\zeta}{Q}}\mathrm{erfc}(-\hat{\tau}\sqrt{Q/\zeta})e^{-\hat{\tau}^2/r} \\ &-\frac{\zeta}{Q}\mathrm{erfc}(-\hat{\tau}/\sqrt{r})e^{-Q\hat{\tau}^2/\zeta}\Bigg]\end{aligned} \tag{2.47}$$

Na figura 2.6 desenhamos $e_g(\alpha)$ e $\zeta(\alpha)$ para ABOn escalar, obtidos pela integração das equações acopladas 2.44, 2.45 e 2.47. Como podemos observar, o erro de generalização converge para $0,88/\alpha$ que é exatamente o mesmo erro assintótico obtido pelo AO. Contudo, com a grande vantagem de não utilizar informações inacessíveis do tipo ρ ou $\bar{\tau}$ — presentes no AO. Além disso, ao identificarmos $\eta(\alpha) = \sqrt{\zeta(\alpha)}$ na equação 2.39, vemos que assintoticamente $\eta(\alpha) \propto 1/\alpha$, que para $\eta(\alpha)$ é justamente uma condição suficiente para uma convergência local de $\hat{\omega}(\alpha)$ a $\bar{\omega}$ [Murata et al 2002, Duda et al 2001]. Com relação ao ajuste tensorial do ABOn tensorial, conclui-se que ele atua na redução do coeficiente c ($e_g = c/\alpha^b$) para aprendizagens sem ruído e com $\bar{\omega}$ fixo.

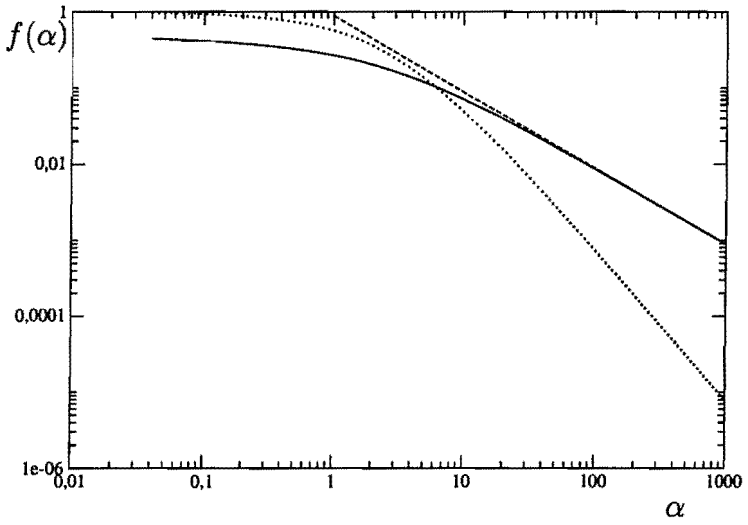

Figura 2.6: Solução numérica para o ABOn escalar. A linha tracejada representa $f(\alpha) = 0,88/\alpha$, a linha cheia $f(\alpha) = e_g(\alpha)$ e a pontilhada $f(\alpha) = \zeta(\alpha)$. Para grandes valores de α temos $e_g(\alpha) \simeq 0,88/\alpha$ e $\zeta \propto \alpha^{-2}$.

2.3 Aprendizagem com dados não-fidedignos.

Finalizamos a seção anterior observando que o ABOn escalar alcança o mesmo desempenho do AO, sendo mais adequado para problemas práticos uma vez que só depende das variáveis $\{\sigma, \boldsymbol{\xi}\}$. Contudo, via de regra os dados utilizados para o treinamento possuem algum tipo de ruído, de forma a não serem totalmente con-

fiáveis. Assim, para realmente conhecermos a sua praticidade, ou aplicabilidade, é necessário sabermos quão robusto é o algoritmo perante a existência de ruídos em D.

Nas próximas subseções estudaremos os principais tipos de ruído encontrados na literatura: ruído aditivo e ruído multiplicativo. Mas antes de prosseguirmos, é necessário distinguirmos o erro de classificação de exemplos não vistos do ângulo entre $\bar{\omega}$ e $\hat{\omega}$, pois mesmo que a rede aprendiz possua $\hat{\omega} = \bar{\omega}$, ainda cometerá erros de predição devido ao ruído. Assim, definimos *erro de predição* e_p como sendo a probabilidade de um exemplo $\xi \notin D$ ser mal classificado; mantemos $e_g = \frac{1}{\pi}\arccos(\rho)$ e também definimos o *erro residual* e_r como sendo o erro causado puramente pelo ruído, ou seja, quando $\hat{\omega} = \bar{\omega}$. Na ausência de ruído teremos $e_p = e_g$ e $e_r = 0$.

2.3.1 Ruído aditivo.

O ruído aditivo pode atuar tanto no vetor de entrada $\boldsymbol{\xi}$ como no professor $\bar{\omega}$, representando uma flutuação nas suas coordenadas. No entanto, devido a natureza da função de transferência $\sigma = \text{sinal}(\bar{\omega} \cdot \boldsymbol{\xi}/\sqrt{N})$ suas atuações são completamente equivalentes. Portanto, trataremos apenas do caso de ruído nas coordenadas de $\bar{\omega}$. Isso significa que ao invés de σ, forneceremos $\tilde{\sigma} = \text{sinal}(\tilde{\omega} \cdot \boldsymbol{\xi}/\sqrt{N})$ para o perceptron aprendiz, com

$$P(\tilde{\omega}^i|\bar{\omega}^i) = \exp\{-(\tilde{\omega}^i - \bar{\omega}^i)^2/2s\}/\sqrt{2\pi s}. \tag{2.48}$$

A magnitude do ruído é caracterizada pelo valor de $s \geqslant 0$; quando maior s maior o nível de ruído. Pela expressão 2.48, também observamos que as entradas $\boldsymbol{\xi}$ muito distantes da superfície de separação[19] recebem pouca influência desse tipo de ruído. Isso implica uma faixa de incerteza em torno da superfície de separação que gera um erro residual.

Pela figura 2.7, normalizando-se $\bar{\omega}$, vemos intuitivamente que esse erro mínimo é dado por[20] $e_r = \frac{1}{\pi}\arccos\epsilon$, sendo $\epsilon = 1/\sqrt{1+s}$. Por sua vez, o erro de predição é dado por $e_p = \frac{1}{\pi}\arccos\epsilon\rho$ [Engel et al 2001].

[19] O hiperplano definido por $\bar{\omega}$.

[20] $e_r = e_p(\hat{\omega} = \bar{\omega}) = \frac{2\theta}{2\pi}$ e $\cos\theta = 1/\sqrt{1+s}$.

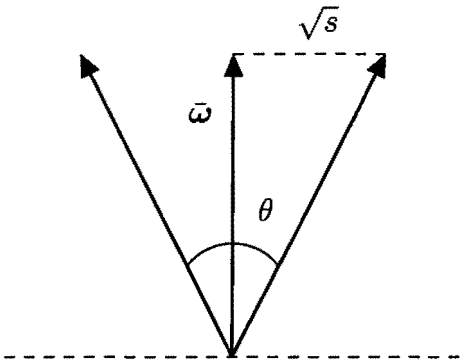

Figura 2.7: Erro residual para o perceptron com ruído aditivo: $e_r = \frac{1}{\pi}\arccos\epsilon$, $\epsilon = 1/\sqrt{1+s}$.

Embora não tenhamos terminado as análises das equações diferenciais envolvidas, podemos perceber por meio das simulações (figuras 2.8 e 2.9) que o algoritmo ABOn tensorial apresenta um desempenho semelhante ao escalar. Além disso é interessante enfatizar que mesmo sem a utilização da informação de existência de ruído, os algoritmos alcançaram o regime $\rho \neq 0$ nas simulações realizadas. Isto é extremamente interessante uma vez que tais algoritmos podem ser vistos como versões práticas (realizáveis) do algoritmo ótimo.

Em [Copelli et all 1997] os autores apresentaram um diagrama de robustez para o problema de aprendizagem com ruído por redes com várias camadas. Embora o diagrama para o ruído aditivo tenha sido construído para redes de paridade[21] e comitê, a comparação é totalmente válida para o número de ramos $K = 1$. Assim, observamos que, a semelhança do AO, os ABOn tensorial e escalar também apresentam tolerância ao ruído aditivo com aprendizagem perfeita ($\rho = 1$) para baixos níveis de ruído.

A diferença apresentada pelos algoritmos em questão no erro de generalização para o caso sem ruído, parece não sobreviver quando aplicamos um ruído aditivo nas sinapses do perceptron professor. Isso é perfeitamente compreensível dado que a diferença é expressada no coeficiente do erro de generalização.

[21] Definimos por redes de paridade qualquer rede que possua uma única camada com K ramos, tendo em cada ramo um perceptron simples com N/K sinapses. A entrada $\boldsymbol{\xi} \in \Re^N$ é divida em K sub-entradas $\boldsymbol{\xi}_i \in \Re^{N/K}$ e a saída é dada por $\sigma = \prod_{i=1}^{K} \text{sinal}(\boldsymbol{\omega}_i \cdot \boldsymbol{\xi}_i)$

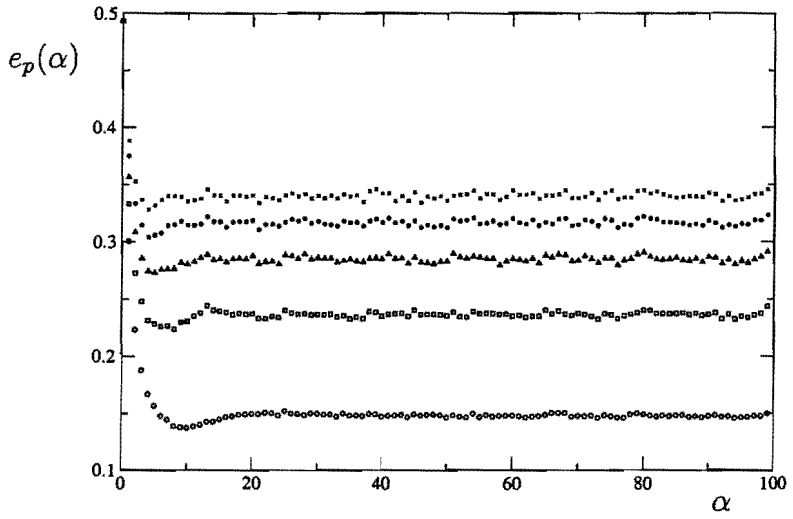

Figura 2.8: Erro de predição para o perceptron com ruído aditivo. Os pontos foram obtidos pelo algoritmo ABOn tensorial, com os seguintes parâmetros: $s = 0,1$ (círculos); $s = 0,3$ (quadrados); $s = 0,5$ (triângulos); $s = 0,7$ (asteriscos) e $s = 0,9$ ($\times$).

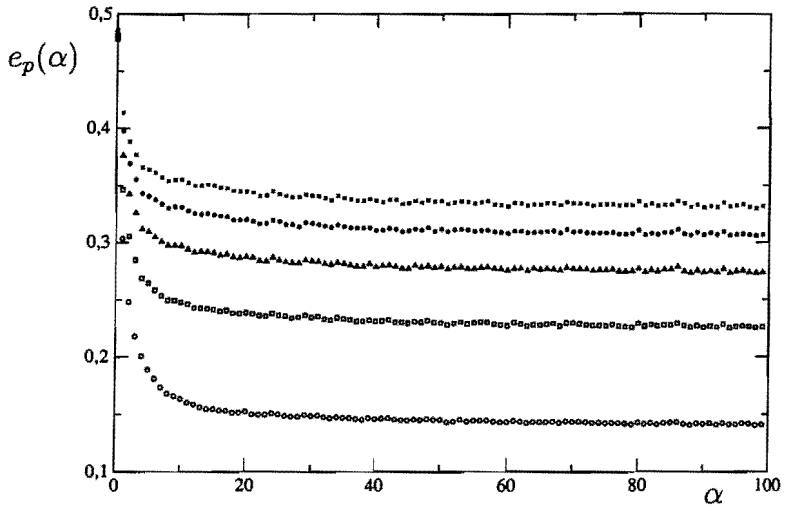

Figura 2.9: Erro de predição para o perceptron com ruído aditivo. Os pontos foram obtidos pelo algoritmo ABOn escalar com os seguintes parâmetros: $s = 0,1$ (círculos); $s = 0,3$ (quadrados); $s = 0,5$ (triângulos); $s = 0,7$ (asteriscos) e $s = 0,9$ ($\times$).

2.3.2 Ruído multiplicativo.

O ruído multiplicativo corrompe a resposta σ independentemente de $\boldsymbol{\xi}$. Neste caso, o conjunto de informações fornecido a rede aprendiz são as entradas $\boldsymbol{\xi}_\mu$ e as saídas corrompidas $\tilde{\sigma}_\mu$ gerados por $\tilde{\sigma}_\mu = \text{sinal}(\epsilon\ \bar{\omega} \cdot \boldsymbol{\xi}_\mu/\sqrt{N})$, com $\epsilon \in \{1, -1\}$ sorteado de acordo com a probabilidade $P(\epsilon = -1) = \chi$, sendo χ constante. Assim, $\tilde{\sigma}_\mu$ é obtido por meio da distribuição[22]:

$$P(\tilde{\sigma}_\mu|\sigma_\mu) = (1-\chi)\delta_k(\tilde{\sigma}_\mu, \sigma_\mu) + \chi\delta_k(\tilde{\sigma}_\mu, -\sigma_\mu). \tag{2.49}$$

Nesse tipo de ruído temos que uma fração χ de exemplos será corrompida, isso significa que em média teremos um erro residual $e_r = \chi$. Pelo mesmo raciocínio podemos chegar facilmente à expressão para o erro de predição[23]:

$$e_p(\alpha) = \chi + \frac{(1-2\chi)}{\pi} \arccos \rho(\alpha). \tag{2.50}$$

Nas figuras 2.10 e 2.11, observamos que os algoritmos ABOn tensorial e escalar apresentam um desempenho equivalente para baixos níveis de ruído. No entanto, ao aumentarmos o nível de ruído percebemos uma sensível superioridade (um desempenho melhor) do ABOn tensorial com relação ao ABOn escalar. Isso ocorre provavelmente devido a termos uma completa independência do ruído multiplicativo com os vetores $\boldsymbol{\xi}_\mu$, de forma a recuperarmos um cenário semelhante ao caso sem ruído.

Numa região de aprendizagem com $\rho \approx 1$, é necessário utilizar vetores que estejam próximos da superfície de separação das classes para se obter um melhoramento no erro de generalização. No ruído aditivo esses vetores são justamente os mais afetados pelo ruído, de forma a não ser possível afinar o conhecimento sobre $\bar{\omega}$. Entretanto, no caso de ruído multiplicativo é possível encontrarmos informações verdadeiras nas vizinhanças da superfície de separação, de forma que é útil guardar informações sobre a correlação das coordenadas de $\hat{\omega}$. Outra forma de ver isso é simplesmente lembrar que o ruído aditivo atua nas coordenadas do perceptron professor $\bar{\omega}$, de forma a danificar seriamente informações sobre sua matriz covariância.

[22] $\delta_k(a,b)$ é a versão discreta da função delta de Dirac, conhecida por *delta de Kronecker*. $\delta_k(a,b) = 1$ para $a = b$ e $\delta_k(a,b) = 0$ caso contrário.

[23] Embora tenhamos apresentado as expressões para e_r e e_p por meio de simples argumentações, ambas (ruído aditivo e multiplicativo) são obtidas sem grandes dificuldades por meio de $e_p = \langle\Theta(-\tilde{\sigma}\ \text{sinal}(\hat{\omega} \cdot \boldsymbol{\xi}))\rangle_{D,\text{ruído}}$.

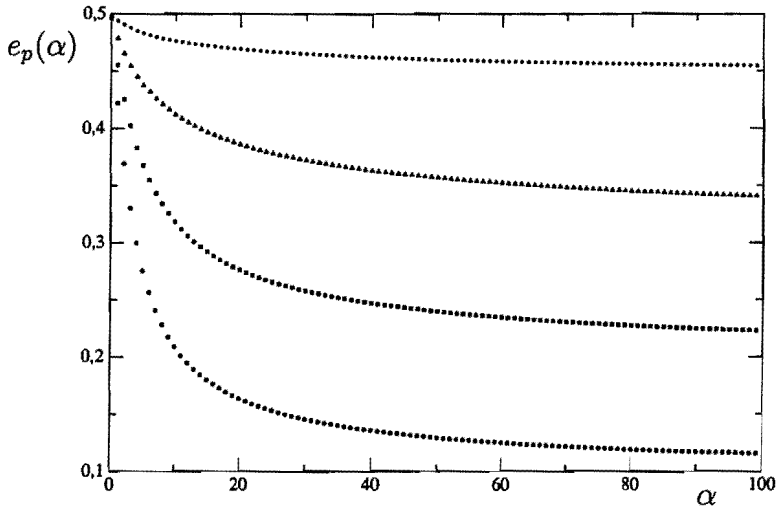

Figura 2.10: Erro de predição para o perceptron com ruído multiplicativo. Os pontos foram obtidos pelo algoritmo ABOn tensorial com os seguintes parâmetros: $\chi = 0,1$ (círculos); $\chi = 0,2$ (quadrados); $\chi = 0,3$ (triângulos); $\chi = 0,4$ (asteriscos).

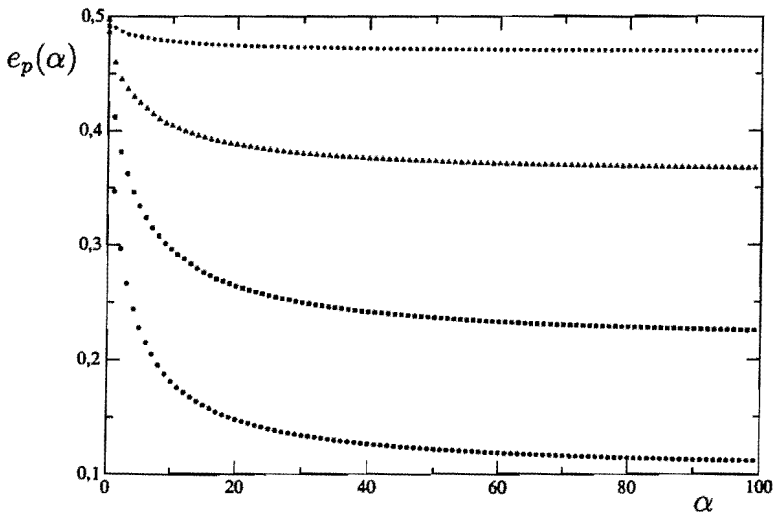

Figura 2.11: Erro de predição para o perceptron com ruído multiplicativo. Os pontos foram obtidos pelo algoritmo ABOn escalar com os seguintes parâmetros: $\chi = 0,1$ (círculos); $\chi = 0,2$ (quadrados); $\chi = 0,3$ (triângulos); $\chi = 0,4$ (asteriscos).

3

Aprendendo conceitos não-estacionários.

Em geral os algoritmos off-line são capazes de aprender mesmo na presença de ruído [Engel et al 2001]. Essa habilidade é facilmente compreendida no ABOff, onde a previsão $\hat{\sigma}$ da rede aprendiz é dada pelo voto majoritário dos candidatos a $\bar{\omega}$, de forma que as informações corrompidas são descartadas uma vez que são minoria[1]. Entretanto, vimos que o mesmo já não é tão verdade nos algoritmos ABOn tensorial e escalar. Em tais algoritmos existe um nível crítico de ruído (menor que $0,5$) para o qual não é possível aprender perfeitamente. Claro que podemos aumentar tal nível adicionando a informação de existência de ruído. Essa adição geralmente é feita por meio de modificações em $P(y|\omega)$ [2], mas isso nos levaria a um outro problema: estimar o nível de ruído em D. De qualquer maneira, podemos dizer que a motivação para a construção de algoritmos on-line devido ao baixo custo de processamento cai consideravelmente com a aquisição de computadores mais potentes, o que em nossos dias tem sido uma empolgante realidade. Isso nos leva à inevitável pergunta: *existe alguma situação na qual os algoritmos on-line realmente devam ser preferidos aos off-line?*

[1] Para o caso de ruído multiplicativo, teremos a aprendizagem de $-\bar{\omega}$ quando a quantidade de informações corrompidas for maior que a quantidade de não-corrompidas.

[2] Por exemplo, fixando β.

Em muitos problemas de aprendizagem a fonte que se deseja estudar muda durante a coleta dos dados utilizados no treinamento. Dependendo da escala de tempo dessa mudança, os dados mais antigos podem se tornar completamente descorrelacionados com o recente estado da fonte, ou seja, inválidos. Nesse cenário, a seqüência de coleta dos dados é uma informação importante para a inferência do estado atual da fonte. De forma que, processar os dados sem levar em consideração tal seqüencia acarreta grandes erros na inferência de novos dados, tanto maior quanto maior a velocidade de mudança da fonte ou conceito. Visto que num treinamento off-line os dados são processados em paralelo, não importando a seqüência da coleta, podemos dizer que a aprendizagem de conceitos que podem mudar no tempo é um problema intrinsecamente on-line. Assim, uma motivação maior para o estudo de algoritmos on-line é a própria *aprendizagem on-line*, ou seja, o processo de aprendizagem contínua e interativa[3].

Em geral podemos classificar os diversos tipos de mudanças de conceito em contínuas ou momentâneas. Dentre essas duas classes existem diversas dinâmicas possíveis para a rede professor $\bar{\omega}$. Contudo, como estamos interessados nas propriedades gerais da aprendizagem por perceptrons nos concentraremos apenas em três casos extremos representando: mudanças aleatórias contínuas, mudanças aleatórias instantâneas e mudanças determinísticas contínuas.

O problema proposto continua sendo classificação binária, mas agora com σ_μ determinado por um professor $\bar{\omega}_\mu$ que muda a cada passo de acordo com a dinâmica

$$\bar{\omega}_{\mu+1} = \left(1 - \frac{1}{N}\Lambda_{\mu+1}\right)\bar{\omega}_\mu + \frac{1}{N}\vartheta_{\mu+1}. \tag{3.1}$$

O parâmetro $\Lambda_{\mu+1}$ controla a norma de $\bar{\omega}_{\mu+1}$ enquanto $\vartheta_{\mu+1} \in \Re^N$ define o seu deslocamento.

É importante salientarmos que embora tenhamos uma expressão para a mudança de $\bar{\omega}$, nenhuma informação sobre tal é fornecida aos algoritmos de aprendizagem. A observação de seus respectivos comportamentos e capacidades de adaptação torna-se

[3]Um bom exemplo de um processo de aprendizagem contínua e interativa é o *jogo da minoria* [Challet et al 1997]. Nesse jogo temos M jogadores que a cada passo devem tomar uma decisão dentre duas possíveis ($\sigma \in \{-1, 1\}$). As informações acessíveis são as escolhas da minoria nos passos anteriores e os ganhadores serão os que escolherem a classe σ da minoria, de forma que $\xi_\mu = (\sigma_{\mu-N}, \sigma_{\mu-N+1}, ..., \sigma_{\mu-1})^T$ e a cada passo μ, cada jogador fornece sua escolha $\hat{\sigma}$ e em seguida recebe o par (ξ_μ, σ_μ) para atualizar seus parâmetros.

então extremamente interessante e informativa para a construção de novos algoritmos adaptativos.

3.1 À deriva.

Neste cenário o vetor $\bar{\omega}$ encontra-se numa caminhada aleatória sobre uma esfera N-dimensional de raio unitário.

Partindo da equação 3.1, teremos essa caminhada ao ajustarmos $\vartheta \sim \mathcal{N}(0, 2D\,I)$ com $D \in \Re$ e I sendo a matriz identidade. Já a normalização $||\bar{\omega}_\mu|| = 1$ é obtida por $\Lambda_{\mu+1} = \bar{\omega}_\mu \cdot \vartheta_{\mu+1} + D$ e $\bar{\omega}_0 \cdot \bar{\omega}_0 = 1$.

A velocidade de deslocamento de $\bar{\omega}$ é regulada por D; quanto maior D maior será o deslocamento por passo e mais difícil será para o perceptron aprendiz $\hat{\omega}$ aprender a regra atual, ou melhor, acompanhar o professor. Isso é claramente percebido pela correlação temporal do professor $\langle \bar{\omega}_\mu \cdot \bar{\omega}_\nu \rangle \propto \exp\{-D|\mu - \nu|\}$ [Vicente et al 1998].

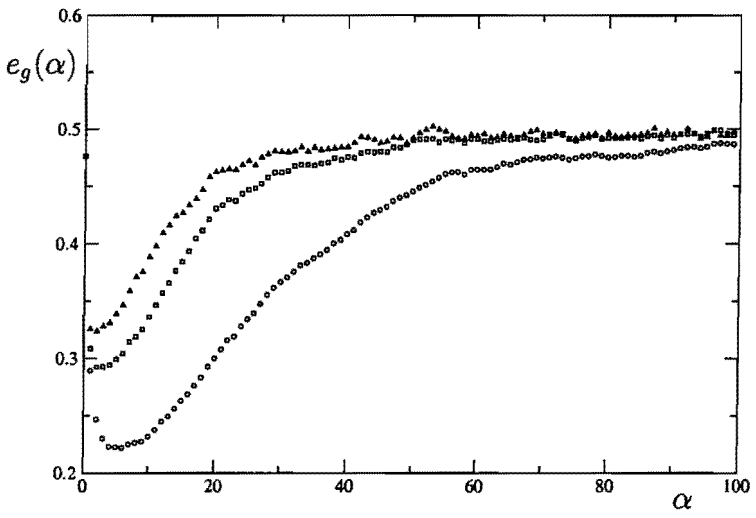

Figura 3.1: Erro de generalização para o perceptron em caminhada aleatória. As curvas foram obtidas pelo algoritmo ABOn escalar com D assumindo os valores: $0,01$ (círculos); $0,03$ (quadrados) e $0,05$ (triângulos).

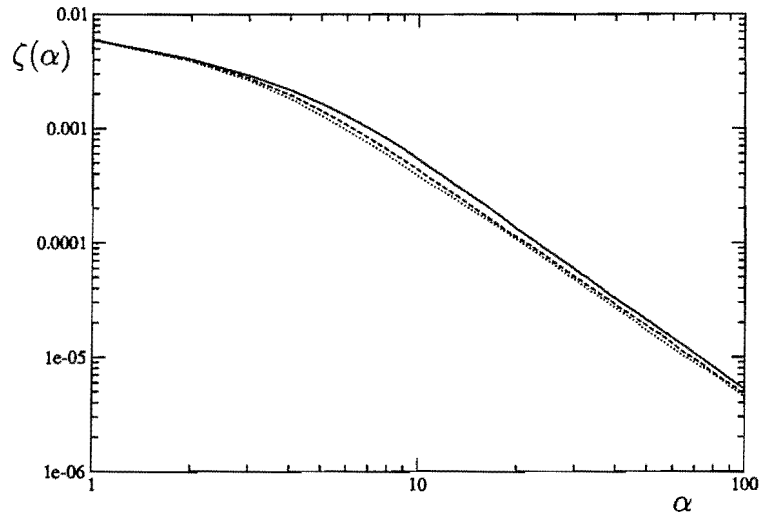

Figura 3.2: $\zeta(\alpha)$ para o perceptron em caminhada aleatória. As curvas foram obtidas pelo algoritmo ABOn escalar com D assumindo os valores: $0,01$ (linha cheia); $0,03$ (linha tracejada) e 0.05 (linha pontilhada).

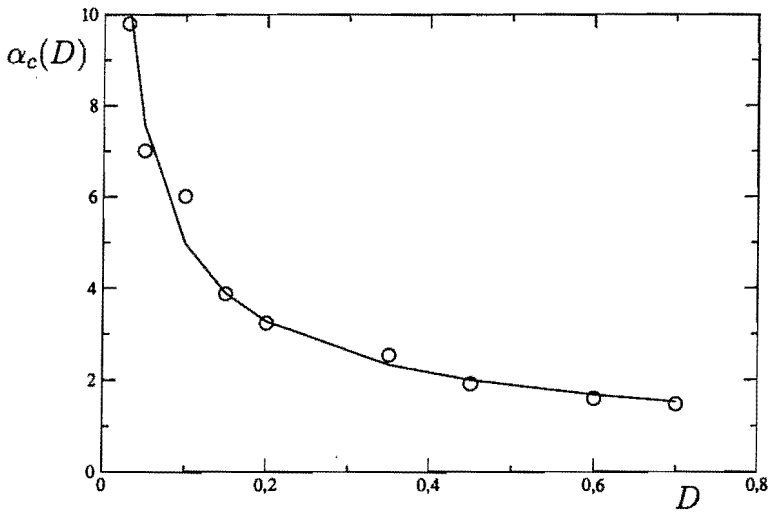

Figura 3.3: $\alpha_c(D)$ para o perceptron em caminhada aleatória. Os pontos foram obtidos pelo algoritmo ABOn escalar e a curva ajustada é $\alpha_c(D) \propto D^{0,61}$ com desvio padrão $\mathcal{O}(10^{-2})$ — no expoente.

Na figura 3.1 vemos que apesar do perceptron aprendiz chegar próximo ao professor no início do processo, existe um ponto $\alpha_c(D)$ a partir do qual ele não consegue

mais acompanhar as mudanças em $\bar{\omega}$, convergindo para $e_g(\alpha) = 0,5$. Isso provavelmente é devido a sempre termos no ABOn escalar uma redução em ζ mesmo para grandes estabilidades $\hat{\tau}$ negativas ($\hat{\tau} \leqslant -1$). Como podemos perceber na figura 2.4, $\hat{\tau} \leqslant -1$ implica grande decréscimo em $\zeta_{\mu+1}$ fazendo com que ele vá a zero muito rápido, o que por sua vez anula a correção $\Delta\omega_{\mu+1}$. De fato, obtivemos $\zeta(\alpha) \xrightarrow{\alpha \gg 1} \alpha^{-n}$ com $n > 2$ em nossas simulações (figura 3.2).

O mesmo comportamento (decréscimo contínuo de x_μ) também ocorre para o ABOn tensorial, contudo, como podemos perceber pela figura 3.4, o algoritmo consegue $\rho(\alpha) \neq 0$, mantendo assim uma perseguição constante ao professor $\bar{\omega}_\mu$. Não há dúvida que tal adaptação é devida ao ajuste tensorial, mas se temos $\mathsf{x}_\mu \to 0$, ou melhor, os elementos de C_μ, como $\Delta\omega_{\mu+1}$ não se anula para tempos grandes?

Para compreendermos essa habilidade não podemos esquecer a essência dos ABOn: a projeção da distribuição posterior numa gaussiana. A largura da gaussiana, dada por C_μ ou ζ_μ, define a incerteza na estimativa $\hat{\omega}_\mu$ e por conseqüência

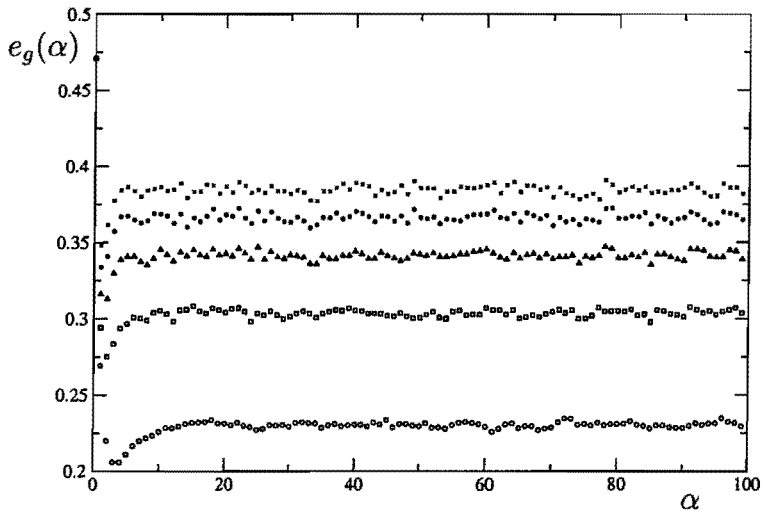

Figura 3.4: Erro de generalização para o perceptron em caminhada aleatória. Os pontos foram obtidas pelo algoritmo ABOn tensorial com: $D = 0,01$ (círculos); $D = 0,03$ (quadrados); $D = 0,05$ (triângulos); $D = 0,07$ (asteriscos) e $D = 0,09$ (x).

a taxa de aprendizagem do algoritmo. Quando $\bar{\omega}_\mu$ é fixo, em ambos algoritmos (tensorial e escalar) $Q \to 1$ e a largura tende a zero, garantindo a convergência

$\hat{\omega}_\mu \rightarrow \bar{\omega}$. Já para o caso de deriva, temos $Q \rightarrow q$, sendo $0 < q < 1$, para o ABOn escalar e $Q \rightarrow 0$ para o ABOn tensorial. Como $\zeta_\mu \rightarrow 0$ ($Q/\zeta_\mu \rightarrow \infty$), a posterior tende a uma distribuição delta, enquanto que para o algoritmo tensorial $Q/\mathrm{x}_\mu \rightarrow$ constante, re-escalando as correções em $\hat{\omega}_\mu$ (veja a figura 3.5).

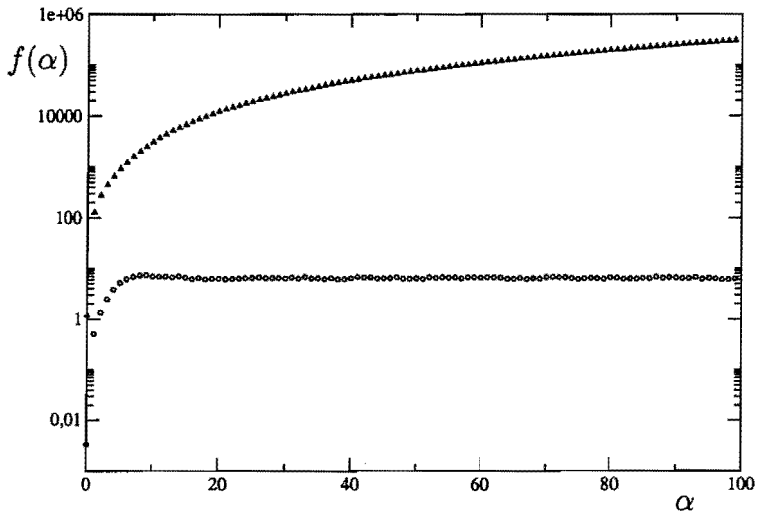

Figura 3.5: Razão entre o tamanho de $\hat{\omega}_\mu$ e uma estimativa da largura da posterior. $f(\alpha) = Q(\alpha)/\mathrm{x}_\mu$ para o ABOn tensorial (círculos) e $f(\alpha) = Q(\alpha)/\zeta_\mu$ para o escalar (triângulos) com $D = 0,1$.

3.2 O teste de Wisconsin.

Como um segundo cenário de mudanças aleatórias veremos o caso de mudanças bruscas momentâneas, definido como *teste de Wisconsin para perceptrons* (TWP) [Vicente et al 1998].

O TWP foi motivado pelo conhecido teste de sorteio de cartas de Wisconsin (Wisconsin card sorting test — WCST) largamente utilizado em diagnósticos sobre disfunções cerebrais [Milner 1963]. O teste consiste em fornecer uma seqüencia de cartas a um paciente que deve classificá-las. Cada carta apresenta um conjunto de figuras de forma a poderem ser separadas em duas categorias, em várias formas (por exemplo, podem ser separadas por cores, número de figuras ou tipo de figuras). Antes de iniciar o teste, o examinador escolhe uma das possíveis regras de classificação. Em

seguida, ele mostra uma carta ao paciente e pergunta sua classificação (o paciente não conhece a regra escolhida pelo examinador), confirmando ou negando a resposta dada. Quando o paciente aprende a regra escolhida inicialmente, o examinador muda a regra sem avisar e continua a perguntar a classificação das cartas. Assim o WCST quantifica a flexibilidade cognitiva, que nada mais é que a habilidade de alterar regras[4] frente a mudanças no ambiente exposto.

Analogamente ao WCST, o TWP consiste em fornecer uma seqüência de pares $\{\boldsymbol{\xi}_\mu; \sigma_\mu\}_{\mu=1}^{\alpha N}$ ordenada temporalmente com o perceptron professor obedecendo:

$$\bar{\omega}_\mu = \sum_{i=1}^{k} \Theta(\mu - \nu_i)\Theta(\nu_{i+1} - \mu)\bar{\omega}_i, \tag{3.2}$$

onde a cada tempo ν_i sorteia-se um novo vetor $\bar{\omega}_i$ de uma dada distribuição.

A figura 3.6 mostra os resultados da simulação do TWP para os algoritmos tensorial e escalar, onde utilizamos $\nu_i = 25(i-1)$. Como pode ser observado, o algoritmo escalar apresenta o mesmo comportamento de um algoritmo puramente *hebbiano*[5], ou seja, apresenta um bom desempenho para primeira regra $\bar{\omega}_0$, mas não consegue acompanhar novos professores. Na verdade, a aprendizagem da segunda regra depende da magnitude de $t_w = \nu_{i+1} - \nu_i$; quanto maior t_w menor $\rho(\alpha)$ — figura 3.7. Contrariamente, a adaptação à segunda regra pelo ABOn tensorial (figura 3.8) parece ser perfeitamente realizável. Isso significa que guardar informações sobre a correlação dos pesos sinápticos é altamente relevante para a aprendizagem de regras que mudam momentaneamente.

Assim como na seção anterior, para o TWP o algoritmo se adapta à nova regra reduzindo $Q(\alpha)$ — figura 3.9. Como o professor $\bar{\omega}_\mu$ permanece constante no intervalo entre ν_i e ν_{i+1}, $Q(\alpha)$ converge para um constante e $\mathrm{x}(\alpha)$ cai semelhante ao caso estacionário. No momento de mudança ν_i observamos uma quebra de derivada tanto em $Q(\alpha)$ e $\zeta(\alpha)$, grandezas totalmente acessíveis. Esse comportamento nos proporciona um promissor detector de mudanças, muito útil em problemas de decisão on-line (automação, ...).

[4]Estabelecidas pelo próprio indivíduo em experiências passadas, ou seja, aprendidas.

[5]Um algoritmo é dito ser hebbiano se a atualização $\Delta\hat{\omega}$ está na direção de $\boldsymbol{\xi}$, ou seja, $\hat{\omega}_{\mu+1} = \hat{\omega}_\mu + cte\ \sigma_\mu \boldsymbol{\xi}_\mu$. O termo hebbiano foi adotado em homenagem ao neuropsicologista Donald O. Hebb, autor do livro "The Organization of Behavior", publicado em 1949, aonde ele apresentou a idéia sobre a aprendizagem sináptica.

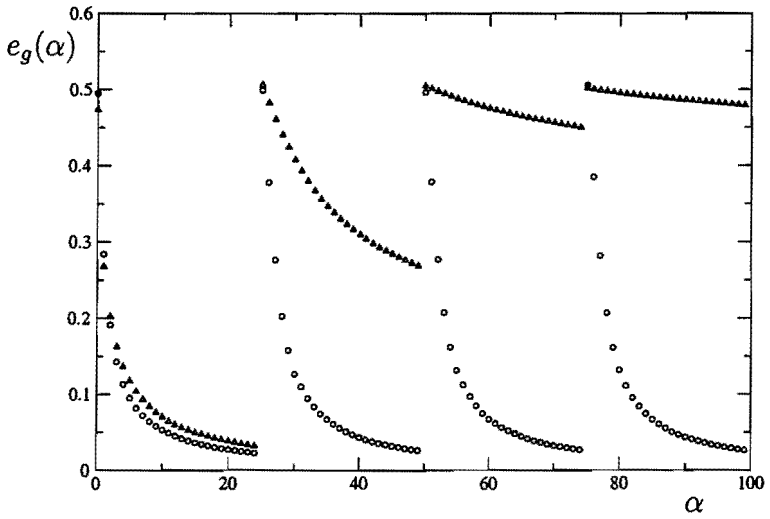

Figura 3.6: Erro de generalização para o perceptron no teste de Wisconsin. Os pontos foram obtidos pelos algoritmos ABOn tensorial (círculos) e escalar (triângulos).

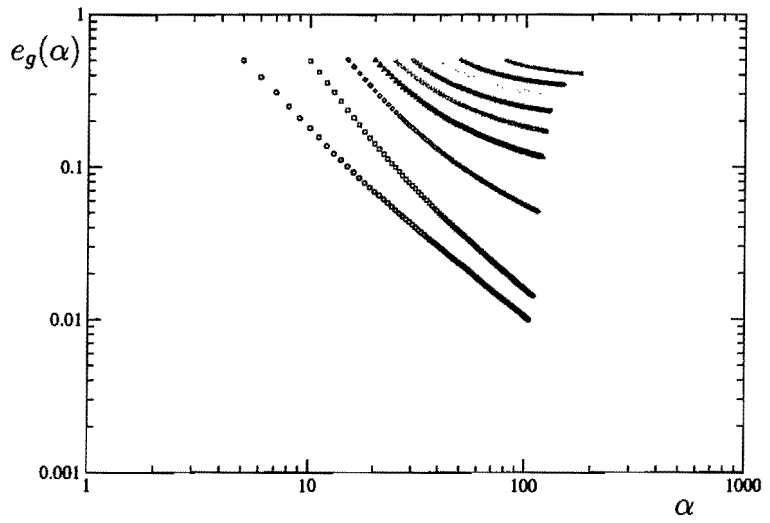

Figura 3.7: Erro de generalização para o ABOn escalar após uma mudança aleatória do professor. Da esquerda para a direita, as curvas foram obtidas com ν_1 igual a $\{5; 10; 15; 20; 25; 30; 40; 50; 80\}$ respectivamente.

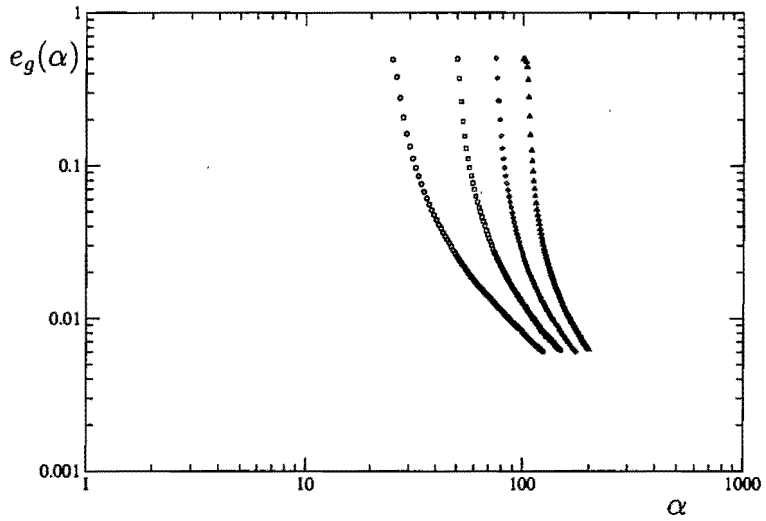

Figura 3.8: Erro de generalização para o ABOn tensorial após uma mudança aleatória do professor. Da esquerda para a direita, as curvas foram obtidas com ν_1 igual a $\{25; 50; 75; 100\}$ respectivamente.

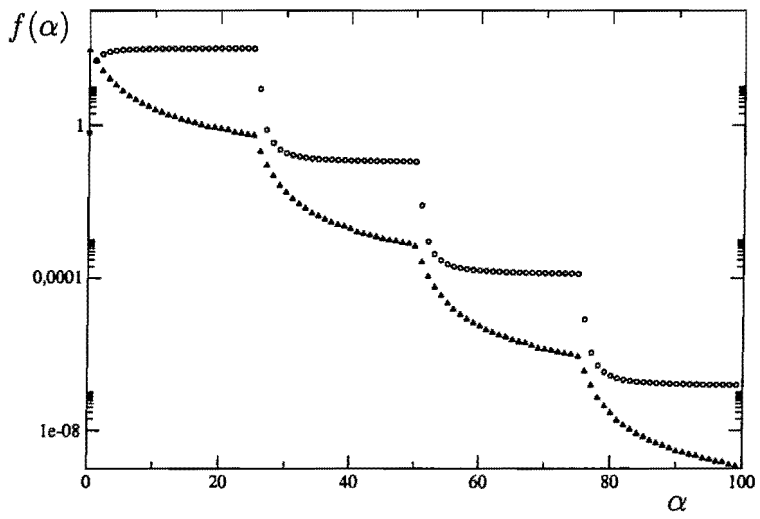

Figura 3.9: $f(\alpha) = Q(\alpha)$ (círculos) e $f(\alpha) = \chi(\alpha)$ (triângulos) para o ABOn tensorial no TWP.

3.3 Fuga.

O pior tipo de mudança acontece quando a cada passo o professor foge o mais rápido possível do aprendiz. Em nosso caso, isso equivale a minimizarmos $\bar{\omega}_{\mu+1} \cdot \hat{\omega}_{\mu}$ sujeito as condições $\bar{\omega}_{\mu+1} \cdot \bar{\omega}_{\mu} = 1 - D/N$ e $||\bar{\omega}_{\mu+1}|| = 1$, que em termos da equação 3.1 significa ajustarmos:

$$\Lambda_{\mu+1} = \frac{D}{N} - ||\vartheta_{\mu+1}||\rho_{\mu}, \tag{3.3}$$

$$\vartheta = -\sqrt{\frac{2D - (D/N)^2}{1 - \rho_{\mu}^2}} \frac{\hat{\omega}_{\mu}}{||\hat{\omega}_{\mu}||}. \tag{3.4}$$

Pela figura 3.10, observamos mais uma vez a incapacidade de aprendizagem do algoritmo ABOn escalar num cenário não-estacionário. Perceba que por não conseguir reajustar a largura da distribuição posterior, o perceptron professor estabiliza na maior distância possível do perceptron aprendiz, ou seja, $\bar{\omega}_{\mu} = -\hat{\omega}_{\mu}/||\hat{\omega}_{\mu}||$.

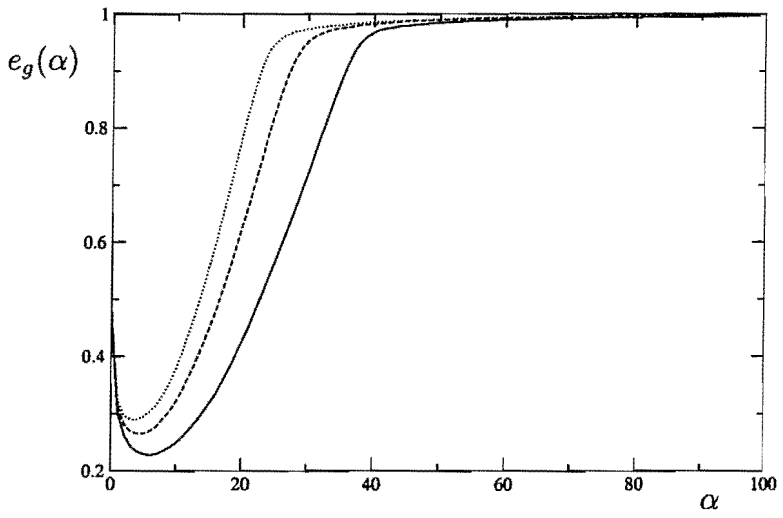

Figura 3.10: Erro de generalização para o perceptron em fuga. As curvas foram obtidas pelo algoritmo ABOn escalar com D assumindo os valores $\{0,01; 0,02; 0,03\}$ nas curvas de baixo para cima.

Em contrapartida, o algoritmo ABOn tensorial (figura 3.11) consegue manter-se constantemente em aprendizagem, de formar que o professor $\bar{\omega}_{\mu}$ não se distancia

muito. Assim, embora ainda haja outras coisas que podem ser analisadas (por exemplo, a dependência do erro mínimo com o parâmetro D no algoritmo tensorial ou o ponto de fadiga do algoritmo escalar) finalizamos esse capítulo ressaltando a importância do comportamento adaptativo do ABOn tensorial, bem como o seu desempenho acima do limite obtido pelo algoritmo ótimo [Kinouchi et al 1992] para o caso estacionário e o conhecimento do porquê da não adaptabilidade do algoritmo escalar.

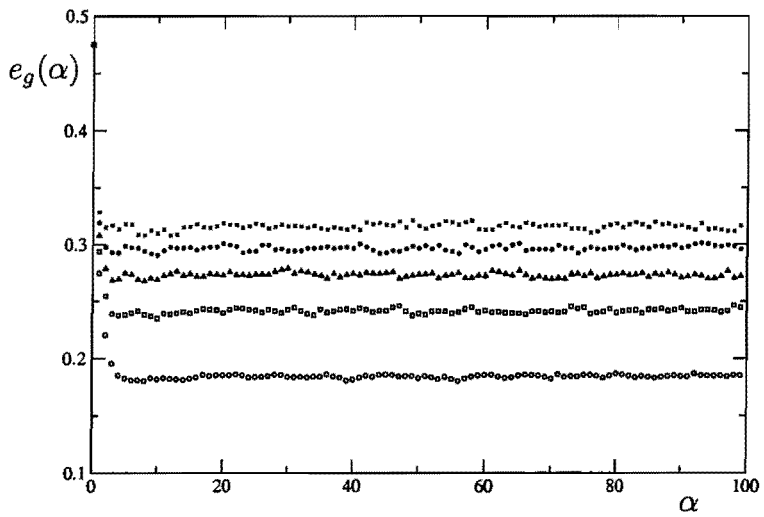

Figura 3.11: Erro de generalização para o perceptron em fuga. As curvas foram obtidas pelo algoritmo ABOn tensorial com D assumindo os valores $\{0,01; 0,03; 0,05; 0,07; 0,09\}$ nas curvas de baixo para cima.

4

Escapando de conceitos velhos.

A habilidade de saber ou decidir até quando deve-se confiar em determinada fonte de informação, ou até mesmo conjunto de informações, é bem conhecida como uma das principais características de um bom aprendiz. Assim, não se espera apenas que o aprendiz aproveite ao máximo cada dado recebido, mas também que este saiba encontrar um ponto de equilíbrio entre segurança[1] e erro de treinamento. Em cenários onde o conceito a ser aprendido muda com o tempo, essa segurança nada mais é que a capacidade de adaptação a novos conceitos.

Para alcançar o equilíbrio entre a adaptação a novos conceitos e o aumento do desempenho assintótico, muitos autores têm lançado mão de algoritmos heurísticos [Widmer et al 1996, Murata et al 2002], nos quais é necessário realizar pré-treinamentos para o ajuste dos parâmetros envolvidos[2]. Entretanto, vimos que o algoritmo ABOn tensorial conseguiu se adaptar às mudanças em $\bar{\omega}_\mu$ sem a adição de nenhuma informação sobre tais. Como ele não foi projetado para cenários não-estacionários, ele sempre diminui a taxa de aprendizagem para garantir uma rápida convergência. Portanto, para permitir correções significativas e manter a diminuição em C_μ, o algoritmo reescala a distribuição posterior por meio da norma $||\hat{\omega}_\mu||$. Apesar disso ser teoricamente suficiente, num sistema real torna-se inviável a medida

[1] Aqui definimos segurança como sendo a probabilidade de manter um bom desempenho em épocas seguintes.

[2] Geralmente os parâmetros estão diretamente relacionados com o tipo de mudança e sua velocidade, ou seja, informações normalmente inacessíveis.

que $x_\mu \to 0$. Logo, a nossa proposta para a obtenção de um novo algoritmo de aprendizado que seja adaptativo frente a novos conceitos, é utilizar o método bayesiano on-line com a adição da informação da possibilidade de mudanças no perceptron professor.

Num cenário não-estacionário, devemos não só estimar o peso sináptico como também o tempo ou validade dos dados fornecidos. Essa validade afeta diretamente η_μ, logo podemos definir nossas variáveis de interesse como sendo o par $\{\omega, \eta\}$, que trocaremos por $\{\omega, \lambda\}$ para não confundirmos as variáveis com suas estimativas (λ é uma variável e η uma constante). Isso significa que a distribuição posterior será projetada para $\mathcal{G}(\omega, \lambda) = G(\omega|\lambda)G(\lambda)$. Além disso, vimos no algoritmo ABOn escalar que $\zeta \equiv \eta$, o que nos permite imaginar que o novo algoritmo é equivalente ao ABOn escalar com a variância dada por uma distribuição *a priori* $G(\lambda)$.

Como família para $\mathcal{G}$ escolhemos a família das distribuições conjugadas:

$$\mathcal{G}(\omega, \lambda) = \frac{e^{-\lambda\|\omega-\hat{\omega}\|^2/2}}{(2\pi/\lambda)^{N/2}} \frac{\lambda^{a-1}e^{-\lambda/b}}{\Gamma(a)b^a}. \tag{4.1}$$

A motivação de 4.1 vem da necessidade de envelhecimento das informações obtidas por meio de D_μ (sem descartá-las) frente a chegada de $y_{\mu+1}$ juntamente com a possibilidade de mudança em $\bar{\omega}$. O descarte de tais informações torna-se desejável apenas quando o novo estado do perceptron professor é completamente descorrelacionado com o seu estado anterior. Para situações tipo deriva, por exemplo, y_μ continua relevante embora dependendo de sua proximidade da superfície de separação não seja mais válido. Essa descrença gradual nos dados pode ser implementada por meio de um multiplicador no potencial de erro, ou seja

$$E_{\mu+1} = V_{\mu+1} + \lambda E_\mu. \tag{4.2}$$

Dessa forma, uma vez que $\lambda < 1$, as contribuições de y_ν serão cada vez menores a medida que 4.2 seja aplicado. Portanto, se a escala temporal das mudanças em $\bar{\omega}$ não é conhecida, podendo até não haver mudanças, torna-se necessário a construção de um algoritmo para o ajuste do multiplicador λ.

Utilizando a equação 4.2 para construir uma distribuição posterior gibbsiana[3] teremos

$$\mathcal{S}_{\mu+1} \propto \exp\{-V_{\mu+1} - \lambda E_\mu\}$$

[3]Como na seção 2.2 e fixando $\beta = 1$.

ou de forma equivalente

$$\mathcal{S}_{\mu+1} \propto \{\mathcal{G}_\mu\}^\lambda \exp\{-V_{\mu+1}\}. \tag{4.3}$$

Perceba que pela expressão 4.3 fica claro que na presença de possibilidade de mudança no perceptron professor, devemos corrigir nossa distruibuição priori, ajustando sua relevância. Em situações não-estacionárias λ deve assumir um valor menor que um, anulando-se para o caso em que o novo estado de $\bar{\omega}$ é completamente descorrelacionado com seu estado anterior. Já em situações estacionárias temos $\lambda = 1$. Se $\mathcal{G}_\mu$ pertencer a uma família de distribuições exponenciais, teremos que λ atuará diretamente na largura da distribuição. Logo, é perfeitamente razoável redefinirmos $\mathcal{G}_\mu$ de forma que sua largura seja dada pelo inverso de λ e, ao invés de termos mais um parâmetro de ajuste para $\mathcal{G}_\mu$, tenhamos o mesmo número de parâmetros do caso estacionário. Com isso, teremos $\lambda < 1$ em situações não-estacionárias e $\lambda \geqslant 1$ em caso contrário.

A minimização de $D_{KL}(S_\mu||\mathcal{G}_\mu)$ é feita sobre os parâmetros $\{\hat{\omega}, a, b\}$ e resulta em[4]

$$\langle \lambda\, \omega^i \rangle_{S_\mu} = \langle \lambda \rangle_{S_\mu} \hat{\omega}^i_\mu \tag{4.4}$$

$$\langle \ln(\lambda) \rangle_{S_\mu} = \Psi(a_\mu) - \ln(b_\mu) \tag{4.5}$$

$$\langle \lambda \rangle_{S_\mu} = a_\mu b_\mu. \tag{4.6}$$

Como $\lambda \geqslant 0$ por definição, temos que o primeiro e o segundo cumulante de $G(\lambda)$ não são independentes entre si. Por exemplo, para a distribuição $G(\lambda)$ escolhida temos que sua média é dada por ab e sua variância ab^2, ou seja, não é possível termos variâncias grandes para pequenos valores da média[5]. Além disso, vemos que variando apenas o valor de b obtemos um conjunto de curvas com a mesma estrutura, enquanto que dependendo do valor de a podemos ter uma curva com convexidade definida[6] ($a = 1$) ou indefinida ($a > 1$) — veja a figura 4.1. De acordo com a expressão 4.1, $G(\lambda = 0)$ só assumirá valor não-nulo se "$a = 1$", por outro lado temos

[4] $\Psi(x) = \mathrm{d}_x \ln \Gamma(x)$, conhecida por função *Digamma*.

[5] Fazendo a mudança $b = c/d$ e $a = d^2/c$ teremos $k_1 = d$ e $k_2 = c$. No entanto é necessário que $a \geqslant 1$, ou seja, $c \leqslant d^2$, de forma a não podemos ter $k_2 > 1$ com $k_1 < 1$.

[6] Uma função é dita ter convexidade definida se ela for estritamente convexa ou côncava e indefinida no caso contrário.

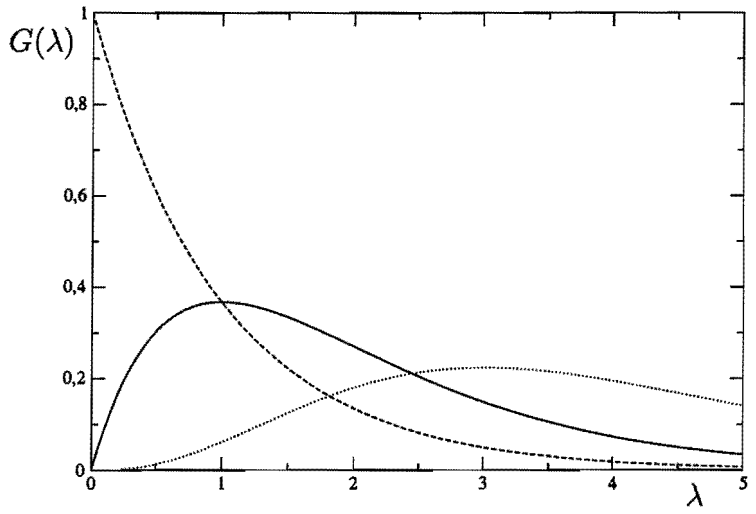

Figura 4.1: $G(\lambda)$ para os seguintes parâmetros $\{a, b\}$: $\{1, 1\}$ (linha tracejada); $\{2, 1\}$ (linha cheia) e $\{4, 1\}$ (linha pontilhada).

$G(\boldsymbol{\omega}|\lambda = 0) = 0$ independente de a. Portanto, como a principal diferença de estrutura devido a a é anulada por $G(\boldsymbol{\omega}|\lambda)$, podemos sem grandes perdas construir o nosso modelo baseado apenas na atualização dos parâmetros $\{\hat{\boldsymbol{\omega}}, b\}$, fixando a e convenientemente escapando da equação 4.5.

Seguindo o mesmo procedimento utilizado no capítulo 2, obtemos as equações:

$$\hat{\boldsymbol{\omega}}_{\mu+1} = \hat{\boldsymbol{\omega}}_{\mu} + \frac{1}{ab_{\mu+1}} \partial_{\hat{\boldsymbol{\omega}}_{\mu}} \ln \langle P(y_{\mu+1}|\boldsymbol{\omega}, \lambda) \rangle_{\mathcal{G}_{\mu}}, \tag{4.7}$$

$$b_{\mu+1} = b_{\mu} + \frac{b_{\mu}^2}{a} \partial_{b_{\mu}} \ln \langle P(y_{t+1}|\boldsymbol{\omega}, \lambda) \rangle_{\mathcal{G}_{\mu}}. \tag{4.8}$$

Lembrando que no caso de interesse $P(y_{t+1}|\boldsymbol{\omega}, \lambda) = F(\sigma_{\mu+1}\boldsymbol{\omega} \cdot \boldsymbol{\xi}_{\mu+1}/\sqrt{N})$ e definindo $\Xi \equiv \langle P(y_{t+1}|\boldsymbol{\omega}, \lambda) \rangle_{\mathcal{G}_{\mu}}$, temos que

$$\begin{aligned}
\Xi &= \int_0^{\infty} \mathrm{d}\lambda \int_{-\infty}^{\infty} \mathrm{d}^N\omega \frac{\mathrm{e}^{-\lambda||\boldsymbol{\omega}-\hat{\boldsymbol{\omega}}_{\mu}||^2/2}}{(2\pi/\lambda)^{N/2}} \frac{\lambda^{a-1}\mathrm{e}^{-\lambda/b_{\mu}}}{\Gamma(a)b_{\mu}^a} \int_{-\infty}^{\infty} \mathrm{d}x\ F(x)\delta\left(x - \sigma_{\mu+1}\boldsymbol{\omega} \cdot \frac{\boldsymbol{\xi}_{\mu+1}}{\sqrt{N}}\right) \\
&= \frac{(2\pi)^{-1/2}}{\Gamma(a)b_{\mu}^a} \int_{-\infty}^{\infty} \mathrm{d}x\ F(x) \int_0^{\infty} \mathrm{d}\lambda\ \lambda^{a-1/2} \exp\{-\lambda[1/b_{\mu} + (x - \tau_{\mu})^2/2]\} \\
&= \frac{\Gamma(a + 1/2)}{\sqrt{2\pi}\Gamma(a)b_{\mu}^a} \int_{-\infty}^{\infty} \mathrm{d}x\ F(x) \left[\frac{1}{b_{\mu}} + \frac{1}{2}(x - \tau_{\mu})^2\right]^{-a-1/2},
\end{aligned} \tag{4.9}$$

onde consideramos $\boldsymbol{\xi}\cdot\boldsymbol{\xi}\to N$ e $\tau_\mu=\sigma_{\mu+1}\hat{\boldsymbol{\omega}}_\mu\cdot\boldsymbol{\xi}_{\mu+1}/\sqrt{N}$. Como $a\geqslant 1$ e fixado num número inteiro, podemos reescrever a última integral em 4.9 em termos da $(a-1)$-ésima derivada com relação a $1/b_\mu$

$$\Xi=\frac{\Gamma(a+1/2)}{\sqrt{2\pi}\Gamma(a)b_\mu^a}\,\frac{(-2)^{a-1}}{(2a-1)!!}\,\frac{\partial^{a-1}}{\partial(b_\mu^{-1})^{a-1}}\int_{-\infty}^{\infty}\mathrm{d}x\;F(x)\left[\frac{1}{b_\mu}+\frac{1}{2}(x-\tau_\mu)^2\right]^{-3/2}$$

que para $F(x)=\Theta(x)$ é facilmente calculada, resultando em

$$\begin{aligned}\Xi &= \frac{\Gamma(a+1/2)}{\sqrt{\pi}\Gamma(a)b_\mu^a}\,\frac{(-2)^{a-1}}{(2a-1)!!}\,\frac{\partial^{a-1}}{\partial(b_\mu^{-1})^{a-1}}\,b_\mu\left[\frac{x}{\sqrt{1+x^2}}\right]_{-\sqrt{b_\mu/2}\tau_\mu}^{\infty}\\ &= \frac{\Gamma(a+1/2)}{\sqrt{\pi}\Gamma(a)b_\mu^a}\,\frac{(-2)^{a-1}}{(2a-1)!!}\,\frac{\partial^{a-1}}{\partial(b_\mu^{-1})^{a-1}}\,b_\mu\left[1+\frac{\sqrt{b_\mu}\tau_\mu}{\sqrt{2+b_\mu\tau_\mu^2}}\right].\end{aligned}\tag{4.10}$$

Mesmo em sistemas não-estacionários esperamos que a velocidade de mudança em $\bar{\boldsymbol{\omega}}_\mu$ não seja tão grande, assim escolhemos $a=2$ por apresentar a menor variância possível em λ em termos de a, com $G(\lambda=0)=0$. Com isso temos

$$\Xi=\frac{1}{2}\left[1+\frac{\sqrt{b_\mu}\tau_\mu}{\sqrt{2+b_\mu\tau_\mu^2}}\left(1+\frac{1}{2+b_\mu\tau_\mu^2}\right)\right]\tag{4.11}$$

que apresenta um comportamento sigmoidal a semelhança do potencial on-line para o caso estacionário. Por fim, aplicamos a última expressão para Ξ em 4.7 e 4.8 e encontramos o nosso novo algoritmo que chamaremos de *ABOn EA* [7]:

$$\hat{\boldsymbol{\omega}}_{\mu+1} = \hat{\boldsymbol{\omega}}_\mu+\frac{\sigma_{\mu+1}\mathcal{F}_\omega}{2b_{\mu+1}}\sqrt{\frac{b_\mu}{N}}\boldsymbol{\xi}_{\mu+1}\tag{4.12}$$

$$b_{\mu+1} = \left(1+\frac{1}{4}\mathcal{F}_b\right)b_\mu\tag{4.13}$$

sendo $\mathcal{F}_\omega=\Xi'/\Xi$, $\mathcal{F}_b=\sqrt{b_\mu}\tau_\mu\Xi'/\Xi$ e $\Xi'=3(2+b_\mu\tau_\mu)^{-5/2}$.

Na figura 4.2 observamos a nova função de modulação $\mathcal{F}_\omega$ e a taxa $b_{\mu+1}/b_\mu$. A grande diferença entre $\mathcal{F}_\omega$ e a função de modulação em 2.36 é o comportamento de sua amplitude para grandes estabilidades negativas ($\sqrt{b_\mu}\tau_\mu<-1$), que decresce para a primeira ($\mathcal{F}_\omega$) e cresce para a segunda ($-\mathrm{d}_x\mathcal{E}(x)$, figura 2.4). Já com relação a equação para a taxa de aprendizagem (o inverso de b_μ), vemos que o novo algoritmo a

[7] Abreviação de "Algoritmo Bayesiano On-line Escalar Adaptativo".

ajusta de acordo como seu desempenho atual (figura 4.2), diminuindo-o na presença de acertos ($\tau_\mu > 0$) e aumentando-o na presença de erros ($\tau_\mu < 0$).

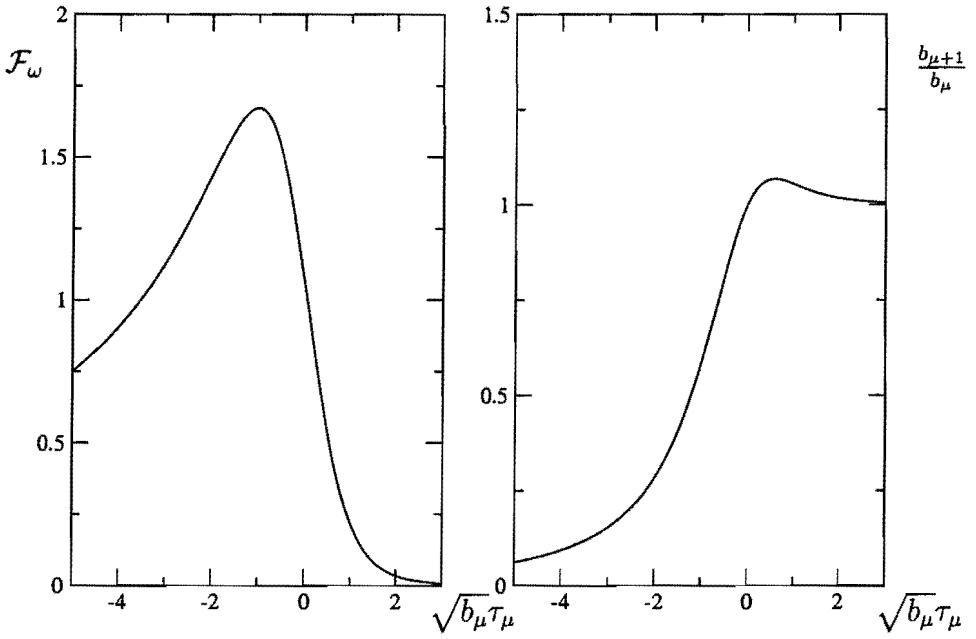

Figura 4.2: Funções de modulação do novo modelo em função da grandeza $\sqrt{b_\mu}\tau_\mu$.

4.1 Conceitos estacionários com e sem ruído.

Embora seja possível construirmos equações diferenciais que representem o algoritmo ABOn EA, nos limitaremos aos resultados obtidos por simulações[8].

Na figura 4.3, observamos que o novo algoritmo também alcança o desempenho ótimo $e_g \approx 0,88/\alpha$ na ausência de ruído ou mudanças no professor, com o mesmo custo computacional do ABOn escalar — o número de operações por dado é de $\mathcal{O}(N)$. Entretanto, apesar de apresentar uma aprendizagem não-nula quando recebe dados com ruído, seu desempenho é inferior aos algoritmos tensorial e escalar (figura 4.4 e 4.5). Uma vez que a única informação sobre a validade dos dados fornecido ao perceptron aprendiz é a possibilidade de envelhecimento, na presença de erros persistentes, ou melhor, erros cometidos quando o aprendiz já deveria ter uma boa

[8]Novamente, fizemos essa escolha para contornar alguns problemas já mencionados no prefácio.

representação do professor, este os interpreta como uma conseqüência de mudança no professor e reduz as correções devidas. Isso pode ser melhor visualizado se nos focalizarmos na regra de Bayes:

$$\mathcal{S}_{\mu+1}(\boldsymbol{\omega},\lambda) = \frac{\mathcal{G}_{\mu}(\boldsymbol{\omega},\lambda)P(y_{\mu+1}|\boldsymbol{\omega})}{\int \mathrm{d}^N\omega' \mathrm{d}\lambda' \mathcal{G}_{\mu}(\boldsymbol{\omega},\lambda)P(y_{\mu+1}|\boldsymbol{\omega})}. \tag{4.14}$$

Quando não temos mudanças em $\bar{\boldsymbol{\omega}}$, a largura da priori $G(\boldsymbol{\omega}|\lambda)$ decresce continuamente de maneira que $\hat{\boldsymbol{\omega}} \to \bar{\boldsymbol{\omega}}$ com $\Delta\hat{\boldsymbol{\omega}} \to 0$. Porém, quando os dados ficam velhos é necessário anular sua influência na posterior. Assim, o algoritmo ABOn EA anula a ação de $G(\boldsymbol{\omega}|\lambda)$ fazendo $b \to 0$ (veja a equação 4.1), ou seja, transformando $G(\boldsymbol{\omega}|\lambda)$ numa distribuição uniforme. Quando o professor é constante, mas os dados não são confiáveis, ou seja, na presença de ruído, não é necessário anular a influência das estimativas anteriores, pois elas continuam válidas. Basta apenas controlar a ação dos dados surpreendentes. Com $G(\boldsymbol{\omega}|\lambda)$ uniforme, as estimativas feitas por meio de 4.14 são baseadas apenas na verossimilhança do último dado $y_{\mu+1}$. Por isso, observamos que o desempenho do novo algoritmo é inferior aos algoritmos escalar e tensorial quando utilizamos dados com ruído, "sem considerar a existência de tal".

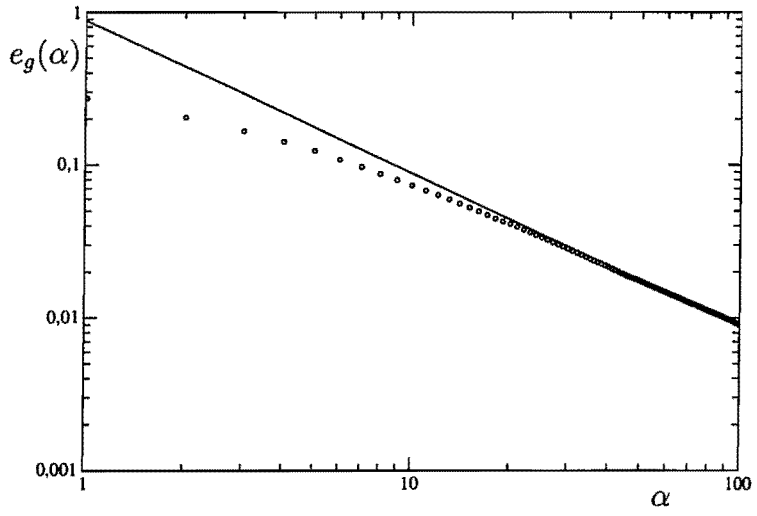

Figura 4.3: Erro de generalização para o perceptron com $\bar{\boldsymbol{\omega}}$ constante. A linha cheia representa a curva $0.88/\alpha$ e os pontos foram obtidos pelo ABOn EA.

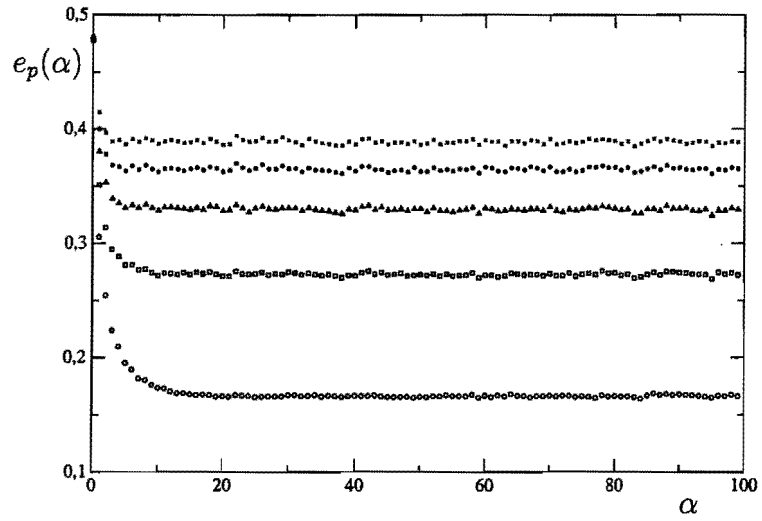

Figura 4.4: Erro de predição para o perceptron com ruído aditivo. Os pontos foram obtidos pelo algoritmo ABOn EA com o seguintes parâmetros: $s = 0,1$ (círculos); $s = 0,3$ (quadrados); $s = 0,5$ (triângulos); $s = 0,7$ (asteriscos); $s = 0$, (x).

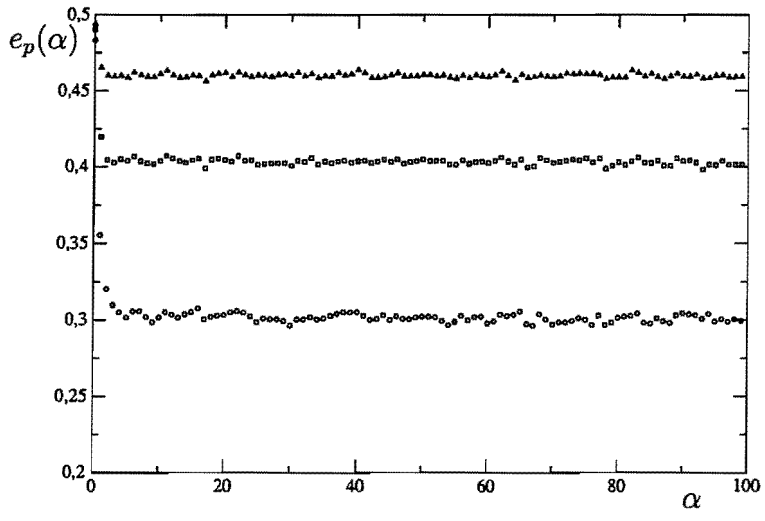

Figura 4.5: Erro de predição para o perceptron com ruído multiplicativo. Os pontos foram obtidos pelo algoritmo ABOn EA com o seguintes níveis de ruído: $\chi = 0,1$ (círculos); $\chi = 0,2$ (quadrados); $\chi = 0,3$ (triângulos).

4.2 Conceitos não-estacionários.

Aplicamos o novo algoritmo nos mesmos casos de mudança em $\bar{\omega}$ do capítulo anterior. Em todos os cenários experimentados (deriva, fuga e TWP) ele apresentou um desempenho semelhante ao ABOn tensorial, que por sua vez se aproxima do desempenho obtido pelo algoritmo AO. Na figura 4.6, apresentamos o erro de generalização assintótico em função do parâmetro D. Para o caso de deriva, o ABOn EA alcançou $e_g \approx 0,36D^{0,24}$ com um desvio padrão $\mathcal{O}(10^{-2})$ em ambos parâmetros de ajuste (coeficiente e expoente). Já para o caso de fuga, encontramos $e_g \approx 0,54D^{0,21}$ com um desvio padrão $\mathcal{O}(10^{-3})$. Comparando esses resultados com os obtidos pelo AO, $e_g \approx 0,45D^{0,33}$ para deriva e $e_g \approx 0,63D^{0,25}$ para fuga [Vicente et al 1998], vemos que o novo algoritmo apresenta um desempenho muito bom em cenários para os quais foi planejado, ou seja, para aprendizagem sem ruído com ou sem mudanças em $\bar{\omega}$. Na figura 4.7 observamos que o ABOn EA consegue facilmente se readaptar durante o TWP, mesmo quando as primeiras mudanças tardam a realizarem-se (figura 4.9). Em oposição a taxa x_μ, b_μ decresce na transição de $\bar{\omega}_\mu$ para $\bar{\omega}_{\mu+1}$ (figura 4.8), voltando a crescer em seguida. Embora exista um pequeno reescalonamento em Q, sua magnitude não é tão grande quanto no ABOn tensorial.

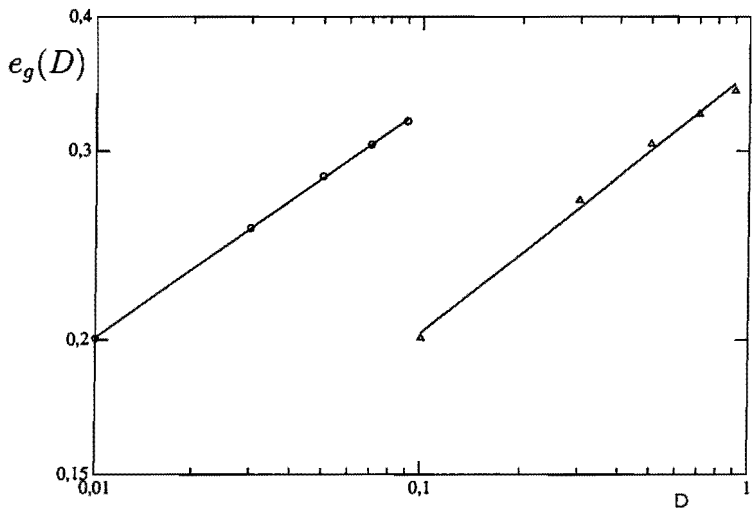

Figura 4.6: Erro de generalização assintótico em função de D para o algoritmo ABOn EA. Os pontos foram obtidos para os seguintes casos: $\bar{\omega}$ à deriva (círculos); $\bar{\omega}$ em fuga (triângulos).

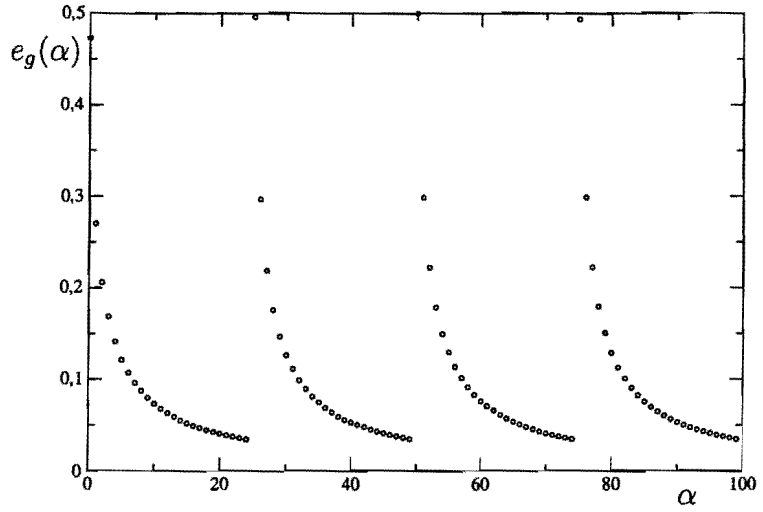

Figura 4.7: Erro de generalização obtido pelo ABOn EA no teste de Wisconsin.

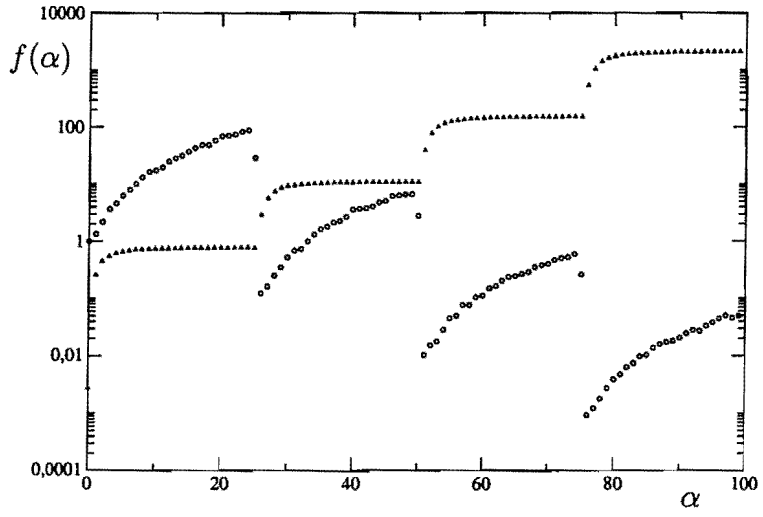

Figura 4.8: $f(\alpha) = Q(\alpha)$ (triângulos) e $f(\alpha) = b(\alpha)$ (círculos) no TWP para o ABOn EA.

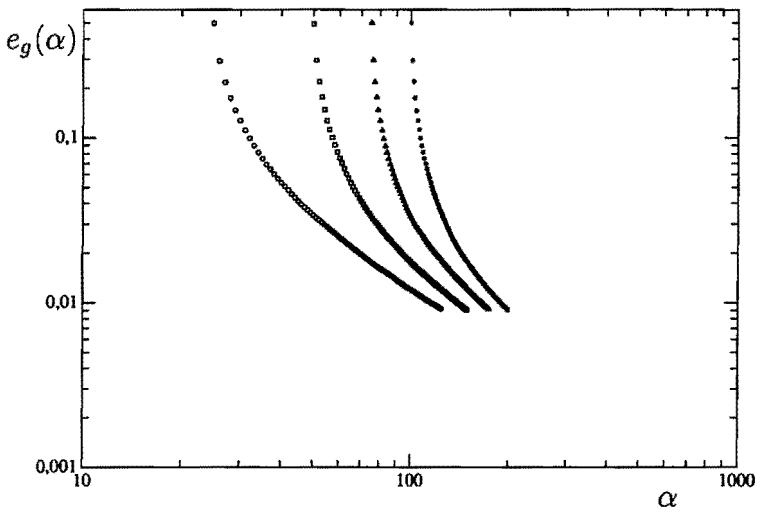

Figura 4.9: Erro de generalização para o ABOn EA após uma mudança aleatória do professor. Da esquerda para a direita as curvas foram obtidas com ν_1 igual a $\{25; 50; 75; 100\}$.

5

Conclusões e perspectivas.

Acreditamos que a maior contribuição desse trabalho não sejam apenas os resultados alcançados pelo novo algoritmo proposto, mas principalmente os conhecimentos sobre algoritmos de aprendizagem sob o ponto de vista bayesiano, adquiridos em paralelo.

Um dos grandes problemas dos algoritmos DG é o ajuste da taxa de aprendizagem, que para não causar flutuações assintóticas na estimativa deve decair após um determinado transiente o qual, por sua vez, depende da forma do potencial de erro. Numa abordagem bayesiana, o próprio algoritmo estima esse transiente por meio da largura da distribuição priori. Isso pode ser claramente percebido nos algoritmos apresentados nesta tese, onde encontramos que a taxa de aprendizagem é diretamente proporcional à variância da distribuição a priori — por exemplo, veja o ABOn escalar: eq. 2.39 e 2.40. Essa variância também depende do potencial de erro, mas não do mesmo potencial V fornecido ao modelador. Na verdade, todo o algoritmo depende do potencial induzido $\mathcal{E}_x = -\ln \langle \exp -\beta V \rangle$.

A transmutação de potenciais também foi observada em algoritmos obtidos pelo método de otimização variacional [Kinouchi et al 1992] — onde o objetivo era encontrar um potencial extremando medidas de desempenho para maximizar a generalização alcançada — bem como a própria relação 2.31 entre o potencial off-line e o seu equivalente off-line, encontrada ao aplicar-se tal método no perceptron [Kinouchi et al 1996]. Isso significa que o ABOn utiliza a mesma forma funcional ótima com relação ao critério de minimização do erro de generalização, o que foi

comprovado no capítulo dois (figuras 2.5 e 2.6). No entanto, as taxas de aprendizagem são ajustadas diferentemente. No AO, a taxa de aprendizagem é dada pela raiz quadrada de $r = 1/\rho^2 - 1$, enquanto que no ABOn é dada pela largura da priori. Claro que para questões práticas o ABOn é mais adequado uma vez que dificilmente se tem acesso a grandezas tipo ρ, mas a essência da taxa de aprendizagem é claramente revelada em r. No início do processo de aprendizagem a correlação entre o perceptron professor e o aprendiz é pequena, ou seja, r é grande. Na medida que $\hat{\omega}$ se aproxima de $\bar{\omega}$, $\rho \to 1$ e $r \to 0$. No ABOn, o aprendiz não tem acesso a ρ e estima sua correlação com o professor baseando-se na grandeza $\langle(\omega - \langle\omega\rangle) \otimes (\omega - \langle\omega\rangle)\rangle_{\omega,D}$. Isso traz uma sensível diferença: o ajuste tensorial.

A atuação de um ajuste tensorial em algoritmos DG, por sua vez, foi observada por Amari [Amari 1998]. Trocando o gradiente $\nabla_{\hat{\omega}}$ pelo seu contravariante $\tilde{\nabla}_{\hat{\omega}} = \mathcal{M}^{-1}\nabla_{\hat{\omega}}$, batizado como *gradiente natural* (GN) e sendo $\mathcal{M}$ o tensor métrico induzido no espaço dos pesos sinápticos ω^i's, Amari mostrou que o algoritmo

$$\hat{\omega}_{\mu+1} = \hat{\omega}_\mu - \eta\mathcal{M}^{-1}\nabla_{\hat{\omega}}\mathcal{E}_{\mu+1}$$

é eficiente segundo o critério de Rao-Cramér, uma vez que $\mathcal{M}$ [1] é definido pela própria matriz de informação de Fisher e η seja ajustado em $1/\mu$.

Apesar do algoritmo GN apresentar um desempenho assintoticamente eficiente dado um potencial ou uma verossimilhança, ele possui o inconveniente da necessidade de inversão de $\mathcal{M}$ [2], além, é claro, do ajuste de η. Assim, vemos mais uma vez a vantagem de baixo custo do algoritmo ABOn com um desempenho assintoticamente eficiente. Já com relação ao método variacional, não podemos simplesmente afirmar que o algoritmo ABOn alcança um erro de generalização inferior aos seus algoritmos, pois o erro assintótico $e_g = 0,88/\alpha$ do AO provavelmente será reduzido ao utilizarmos o método variacional com o gradiente natural. Essa redução do e_g nos algoritmos tensoriais acontece devido à isotropia do espaço, criada por meio do tensor métrico [Amari et al 1998], ou no caso, o seu equivalente C^{-1}. Assim, podemos dizer que o algoritmo ABOn tensorial seria o equivalente prático do AO tensorial, uma vez que apresenta desempenho e comportamento equivalentes. No entanto, é importante salientarmos que a capacidade adaptativa do AO tensorial não estaria na correção do gradiente, como acontece no ABOn tensorial, mas na taxa r.

[1] Diferentemente da matriz H do algoritmo de Newton (equação 2.12), $\mathcal{M}$ é sempre positiva definida.

[2] O custo computacional para a inversão de uma matriz $N \times N$ é de $\mathcal{O}(N^3)$.

Para adquirir uma capacidade adaptativa a novos conceitos, muitos autores propõem a manipulação heurística da taxa de aprendizagem [Widmer et al 1996, Koychev 2000]. Contudo, isso geralmente implica um ajuste *ad hoc* de parâmetros relacionados com a escala de tempo da validade dos conceitos. Esse ajuste pode ser feito tanto por um pré-treinamento off-line [Murata et al 2002] ou pela fixação de uma precisão e uma tolerância ao erro instantâneo [Gama et al 2004], ou seja, $\delta_1 \leqslant e_\mu \leqslant \delta_2$. Devido à diversidade de propostas e suas respectivas inconveniências[3], podemos perceber que a obtenção de informações sobre a dinâmica de $\bar{\omega}$ para uma boa escolha de η (e demais parâmetros relacionados), ou melhor, como escolher bons valores para tais parâmetros, é uma questão ainda não satisfatoriamente respondida e ainda requer bastante cuidado e experimentação. Assim, ao invés de nos lançarmos à procura de conhecimentos a priori para escolha desses parâmetros, nos ativemos à questão: *como inserir a informação de mudanças de conceitos, numa maneira mais natural, em algoritmos bayesianos?* Embora seja controverso definir o que seria "mais natural", apenas decidimos que, se era para aplicarmos conhecimentos *a priori* – antes de iniciar o treinamento on-line – então que o fizéssemos *a la* Bayes, uma vez que os algoritmos com melhores desempenhos, tanto off-line quanto on-line, são bayesianos.

Sabemos que no limite em que o número de elementos em D tende a infinito, a distribuição posterior tende a uma gaussiana centrada na estimativa que maximiza a verossimilhança e com uma largura dada pela derivada segunda do logaritmo da verossimilhança [Walker 1969], para qualquer distribuição a priori absolutamente contínua. Assim, visto que a exigência de tal limite é dificilmente cumprida em problemas práticos, a abordagem bayesiana torna-se bastante atraente e a escolha de gaussianas como a classe de famílias $\mathcal{G}$, no ABOn, intuitiva, principalmente se observarmos a equação 2.9 para a atualização de C_μ. Para conceitos estacionários, vimos que no decorrer do processo de aprendizagem a distribuição posterior desloca-se e contrai-se, tendendo a uma distribuição tipo delta. No entanto, quando $\bar{\omega}$ não permanece constante $||\bar{\omega}_{\mu+1} - \bar{\omega}_\mu||$ eventualmente pode ser grande (muito maior que $|C_\mu|$), de forma a tornar necessário o alargamento da distribuição posterior. Para contornar esse problema, vimos que o ABOn tensorial reescala os pesos sinápticos. Isto está intuitivamente representado na figura 5.1. Perceba que a largura da distri-

[3]Ter que realizar um pré-treinamento off-line para depois aplicarmos um outro on-line ou como escolher δ_1 e δ_2.

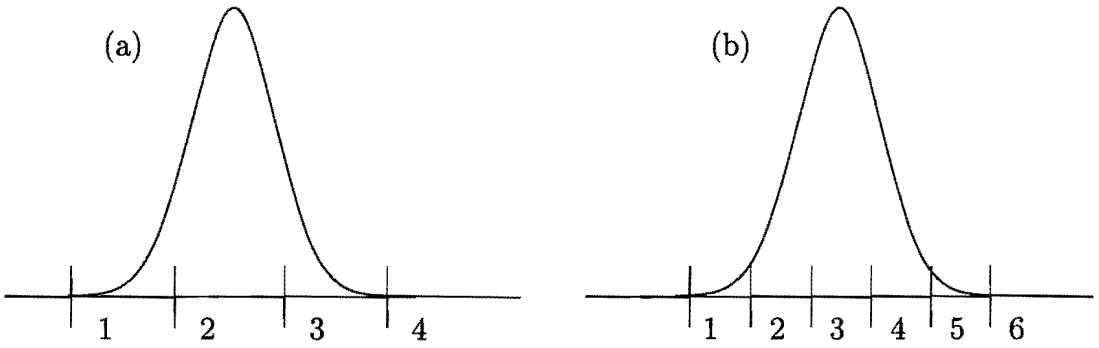

Figura 5.1: Reescalonamento no ABOn tensorial.

buição da figura 5.1(a) é a mesma da 5.1(b), mas as réguas utilizadas são diferentes, de forma que a distribuição em 5.1(b) torna-se mais larga. Assim, compreendendo o problema de envelhecimento nos ABOn's, propusemos uma nova classe de distribuições $\mathcal{G}(\omega, \lambda)$ para representar o perceptron aprendiz, sendo λ a variável relacionada a dinâmica do perceptron professor. Quando o professor sofre uma alteração, o novo dado $y_{\mu+1}$ é completamente válido, ao contrário de $\mathcal{G}_\mu$, ou seja, numa abordagem bayesiana fica claro que:

- quando duvidamos da fidelidade de $y_{\mu+1}$, modificamos $V(y_{\mu+1}; \omega)$;
- quando duvidamos da validade de $\bar{\omega}_\mu$, modificamos $\mathcal{G}_\mu$.

Para compreendermos um pouco mais o efeito de $\mathcal{G} = G(\omega, \lambda)$ dado por 4.1, calculemos $G(\omega) = \int G(\omega, \lambda) \mathrm{d}\lambda$:

$$G(\omega) = \frac{\Gamma(a + 1/2)}{\Gamma(a)\sqrt{2\pi/b}} \left(1 + \frac{b}{2}||\omega - \hat{\omega}||^2\right)^{-a-1/2}. \tag{5.1}$$

A equação 5.1 nos mostra que a distribuição $G(\omega)$ apresenta uma cauda de decaimento algébrico, ou seja, uma longa cauda propícia para grandes saltos ($\hat{\omega}_\mu \rightarrow \hat{\omega}_{\mu+1}$). Isso significa que o perceptron aprendiz está pronto para desistir de suas convicções sobre $\bar{\omega}_\mu$ frente a novas evidências $y_{\mu+1}$.

Por fim, finalizamos esse trabalho ressaltando mais uma vez a semelhança dos desempenhos alcançados entre o ABOn EA e o AO, bem como o seu baixo custo e, principalmente, a inferência da escala de mudanças do perceptron professor pelo aprendiz sem prejudicar o seu desempenho em situações estacionárias. Claro que

ainda há muito o que fazer, como por exemplo preparar o algoritmo para tratar dados não-fidedignos ou até mesmo aplicar o mesmo método em problemas cuja superfície de separação seja não-linear[4], bem como compreender (dinâmica da métrica induzida nos espaços das sinapses). Por isso esperamos que esse trabalho sirva como motivador para o estudo da teoria bayesiana das probabilidades e sua aplicação no estudo da inteligência artificial.

[4]Perceptrons com várias camadas ou máquinas núcleo.

A

Identidades e funções matemáticas

Nesta capítulo apresentamos algumas definições das principais funções fundamentais utilizadas no desenvolvimento dessa tese, bem como as demonstrações de algumas identidades igualmente cruciais para a obtenção das equações dos modelos apresentados.

A.1 Algumas funções matemáticas.

Delta de Dirac

A "função" Delta de Dirac é representada por $\delta(x - x_0)$ e definida como

$$\delta(x - x_0) = \begin{cases} 0 & \forall \quad x \neq x_0 \\ \infty & \text{em} \quad x = x_0 \end{cases} \tag{A.1}$$

com $\int_{-\infty}^{+\infty} \delta(x - x_0)\mathrm{d}x = 1$. Além disso, a função Delta de Dirac também pode ser definida como um funcional por meio da identidade:

$$\int_I \mathrm{d}x f(x)\delta(x - x_0) = \begin{cases} f(x_0) & \text{se} \quad x_0 \in I \\ 0 & \text{se} \quad x_0 \notin I. \end{cases} \tag{A.2}$$

Por fim, apresentamos abaixo a representação de Fourier para a Delta de Dirac, utilizada nos cálculos dessa tese.

$$\delta(x - x_0) = \frac{1}{2\pi} \int_{\infty}^{\infty} \mathrm{d}k \; e^{ik(x - x_0)} \tag{A.3}$$

Função degrau (Heaviside)

A função degrau é definida como

$$\Theta(x) = \begin{cases} 1 & \text{para } x \geq 0 \\ 0 & \text{de outra maneira} \end{cases} \tag{A.4}$$

ou por sua representação integral $\Theta(x - x_0) = \int_{x_0}^{\infty} \mathrm{d}y\ \delta(x - y)$.

Função erro complementar (erfc)

A função erfc(x) é definida como sendo o complemento da função erro erf(x)

$$\begin{aligned} \operatorname{erfc}(x) &= 1 - \operatorname{erf}(x) && \text{(A.5)} \\ &= 1 - \frac{2}{\sqrt{\pi}} \int_0^x \mathrm{d}y\ \exp -y^2 && \text{(A.6)} \\ &= \frac{2}{\sqrt{\pi}} \int_x^{\infty} \mathrm{d}y\ \exp -y^2 && \text{(A.7)} \end{aligned}$$

e, por conseqüência, sua derivada igual a gaussiana $2e^{-x^2}/\sqrt{\pi}$.

A.2 Distribuição de probabilidades dos campos sinápticos.

Para o problema de classificação binária, o rótulo σ_μ para um dado exemplo $\boldsymbol{\xi}_\mu$ depende simplesmente do sinal do produto escalar de $\boldsymbol{\xi}_\mu$ com o professor $\bar{\boldsymbol{\omega}}$, isso significa que podemos trocar o conjunto de variáveis $\{\boldsymbol{\xi}_\mu, \hat{\boldsymbol{\omega}}, \bar{\boldsymbol{\omega}}\}$ por $\{\boldsymbol{\xi}_\mu, \hat{\tau}_\mu, \bar{\tau}_\mu\}$, sendo $\hat{\tau}_\mu$ definido como o campo sináptico do aprendiz e $\bar{\tau}_\mu$ como o campo sináptico do professor.

$$\begin{cases} \hat{\tau}_\mu &= \hat{\boldsymbol{\omega}} \cdot \boldsymbol{\xi}_\mu / ||\hat{\boldsymbol{\omega}}|| \\ \bar{\tau}_\mu &= \bar{\boldsymbol{\omega}} \cdot \boldsymbol{\xi}_\mu / ||\bar{\boldsymbol{\omega}}|| \end{cases} \tag{A.8}$$

Com essa mudança de variáveis podemos definir a distribuição conjunta dos campos sinápticos como:

$$P(\hat{\tau}, \bar{\tau}) = \left\langle \delta\left(\hat{\tau} - \frac{\hat{\boldsymbol{\omega}} \cdot \boldsymbol{\xi}}{||\hat{\boldsymbol{\omega}}||}\right) \delta\left(\bar{\tau} - \frac{\bar{\boldsymbol{\omega}} \cdot \boldsymbol{\xi}}{||\bar{\boldsymbol{\omega}}||}\right) \right\rangle_{\boldsymbol{\xi}}, \tag{A.9}$$

com $\langle\ \rangle_{\boldsymbol{\xi}}$ significando uma média com respeito a $\boldsymbol{\xi}$.

Utilizando a representação de Fourier para a Delta de Dirac:

$$P(\hat{\tau},\bar{\tau}) = \frac{1}{(2\pi)^2}\left\langle \iint \mathrm{d}m\mathrm{d}n\ e^{im\hat{\tau}+in\bar{\tau}} \exp\{-i\left(m\frac{\hat{\omega}}{||\hat{\omega}||}+n\frac{\bar{\omega}}{||\bar{\omega}||}\right)\cdot\boldsymbol{\xi}\}\right\rangle_{\boldsymbol{\xi}};$$

e supondo uma distribuição gaussiana $P(\xi_j) = \frac{1}{\sqrt{2\pi}}e^{-\xi_j^2/2}$, encontramos

$$P(\hat{\tau},\bar{\tau}) = \iiint \frac{\mathrm{d}m\mathrm{d}n\mathrm{d}^N\xi}{(2\pi)^{2+N/2}}\ e^{im\hat{\tau}+in\bar{\tau}} \exp\{-i\left(m\frac{\hat{\omega}}{||\hat{\omega}||}+n\frac{\bar{\omega}}{||\bar{\omega}||}\right)\cdot\boldsymbol{\xi}-\frac{1}{2}\boldsymbol{\xi}\cdot\boldsymbol{\xi}\}$$

, ou seja,

$$P(\hat{\tau},\bar{\tau}) = \iint \frac{\mathrm{d}m\mathrm{d}n}{(2\pi)^2}\ \exp\left\{im\hat{\tau}+in\bar{\tau}-\frac{1}{2}\left(m\frac{\hat{\omega}}{||\hat{\omega}||}+n\frac{\bar{\omega}}{||\bar{\omega}||}\right)\cdot\left(m\frac{\hat{\omega}}{||\hat{\omega}||}+n\frac{\bar{\omega}}{||\bar{\omega}||}\right)\right\}$$

A integração em m e n da última expressão se resumem em simples integrais gaussianas e resulta na distribuição conjunta:

$$P(\hat{\tau},\bar{\tau}) = \frac{\exp\{-(\bar{\tau}-\rho\hat{\tau})^2/2(1-\rho^2)\}}{\sqrt{2\pi(1-\rho^2)}}\ \frac{\exp\{-\hat{\tau}^2/2\}}{\sqrt{2\pi}}. \tag{A.10}$$

Onde foi definido $\rho = \hat{\omega}\cdot\bar{\omega}/||\hat{\omega}||\ ||\bar{\omega}||$.

A distribuição A.10 pode também ser vista como o produto de duas outras distribuições em acordo com a regra de Bayes $P(\hat{\tau},\bar{\tau}) = P(\bar{\tau}|\hat{\tau})P(\hat{\tau})$, sendo

$$P(\bar{\tau}|\hat{\tau}) = \frac{\exp\{-(\bar{\tau}-\rho\hat{\tau})^2/2(1-\rho^2)\}}{\sqrt{2\pi(1-\rho^2)}}, \tag{A.11}$$

$$P(\hat{\tau}) = \frac{\exp\{-\hat{\tau}^2/2\}}{\sqrt{2\pi}}. \tag{A.12}$$

Além disso, observamos que a correlação ρ é auto-mediante, ou seja, $\langle\rho\rangle_{\hat{\tau},\bar{\tau}} = \rho$ [1].

Por meio da regra de Bayes, podemos também encontrar as distribuições $P(\sigma,\hat{\tau})$ e $P(\sigma,\hat{\tau},\bar{\tau})$. Para isso lembremos que $P(\sigma,\hat{\tau}) = P(\sigma|\hat{\tau})P(\hat{\tau})$ e, que por sua vez, $P(\sigma|\hat{\tau}) = \int \mathrm{d}\bar{\tau}P(\sigma,\bar{\tau}|\hat{\tau})$ ou $P(\sigma|\hat{\tau}) = \int \mathrm{d}\bar{\tau}P(\sigma|\hat{\tau},\bar{\tau})P(\bar{\tau}|\hat{\tau})$. Uma vez que $\sigma = \mathcal{F}(\bar{\tau})$, podemos sem grandes perdas considerar $P(\sigma|\hat{\tau},\bar{\tau}) = P(\sigma|\bar{\tau})$, embora em geral $P(\sigma) \neq P(\sigma|\hat{\tau})$. Isso significa que o conhecimento de $\bar{\tau}$ torna $\hat{\tau}$ desnecessário para a

[1] Isso devido a termos $\langle\hat{\tau}\bar{\tau}\rangle_{\hat{\tau}\bar{\tau}} = \rho$.

observação de σ. Dessa forma $P(\sigma|\hat{\tau}) = \int \mathrm{d}\bar{\tau} P(\sigma|\bar{\tau})P(\bar{\tau}|\hat{\tau})$, que para o perceptron booleano[2] transforma-se em

$$P(\sigma|\hat{\tau}) = \int_{-\infty}^{\infty} \mathrm{d}\bar{\tau}\Theta(\sigma\bar{\tau})\frac{\exp\{-(\bar{\tau}-\rho\hat{\tau})^2/2(1-\rho^2)\}}{\sqrt{2\pi(1-\rho^2)}} = \frac{1}{2}\mathrm{erfc}\left(\frac{-\sigma\rho\hat{\tau}}{\sqrt{2-2\rho^2}}\right). \tag{A.13}$$

Por sua vez, temos que

$$P(\sigma,\hat{\tau},\bar{\tau}) = P(\sigma|\hat{\tau},\bar{\tau})P(\hat{\tau},\bar{\tau}) = P(\sigma|\bar{\tau})P(\bar{\tau}|\hat{\tau})P(\hat{\tau}),$$

que é equivalente a

$$P(\sigma,\hat{\tau},\bar{\tau}) = P(\bar{\tau}|\sigma,\hat{\tau})P(\sigma,\hat{\tau}) = P(\bar{\tau}|\sigma,\hat{\tau})P(\sigma|\hat{\tau})P(\hat{\tau}),$$

ou seja, $P(\bar{\tau}|\sigma,\hat{\tau}) = P(\bar{\tau}|\hat{\tau})P(\sigma|\bar{\tau})/P(\sigma|\hat{\tau})$. Logo,

$$P(\bar{\tau}|\sigma,\hat{\tau}) = \frac{\Theta(\sigma\bar{\tau})\exp\{-(\bar{\tau}-\rho\hat{\tau})^2/2(1-\rho^2)\}}{\mathrm{erfc}\left(\frac{-\sigma\rho\hat{\tau}}{\sqrt{2-2\rho^2}}\right)\sqrt{\pi(1-\rho^2)/2}}. \tag{A.14}$$

Para obtermos $P(\sigma,\hat{\tau})$ e $P(\sigma,\hat{\tau},\bar{\tau})$ basta usarmos a regra de Bayes: $P(\sigma,\hat{\tau}) = P(\sigma|\hat{\tau})P(\hat{\tau})$ e $P(\sigma,\hat{\tau},\bar{\tau}) = P(\hat{\tau}|\sigma,\bar{\tau})P(\sigma,\bar{\tau})$.

A.3 Erro de generalização para o perceptron.

O erro de generalização é definido como sendo a probabilidade de um exemplo qualquer (pertencente ao espaço dos exemplos) ser mal-classificado por um dado estado da máquina aprendiz. Para um perceptron isso é equivalente a calcular a região do espaço de versões que possui sinal$(\hat{\omega}\cdot\xi_\mu)$ diferente de σ_μ, ou seja, diferente de sinal$(\bar{\omega}\cdot\xi_\mu)$, ou seja,

$$\mathcal{E}_g = \iint_{-\infty}^{\infty} P(\bar{\tau},\hat{\tau})\Theta(-\bar{\tau}\hat{\tau})\mathrm{d}\bar{\tau}\mathrm{d}\hat{\tau}, \tag{A.15}$$

que pode ser reescrito como

$$\mathcal{E}_g = \int_0^{\infty}\mathrm{d}\bar{\tau}\int_{-\infty}^{0}\mathrm{d}\hat{\tau}P(\bar{\tau},\hat{\tau}) + \int_{-\infty}^{0}\mathrm{d}\bar{\tau}\int_0^{\infty}\mathrm{d}\hat{\tau}P(\bar{\tau},\hat{\tau}). \tag{A.16}$$

[2]No perceptron booleano temos que $\sigma = \mathrm{sinal}(\bar{\tau})$, o que implica em $P(\sigma|\bar{\tau}) = \Theta(\sigma\bar{\tau})$.

Visto que $P(\bar{\tau},\hat{\tau})$ é invariante na troca[3] $\bar{\tau} \leftrightarrows \hat{\tau}$, temos

$$\mathcal{E}_g = 2\int_0^\infty \mathrm{d}\bar{\tau}\int_{-\infty}^0 \mathrm{d}\hat{\tau}\,P(\bar{\tau},\hat{\tau}) = 2\int_0^\infty \mathrm{d}\bar{\tau}\int_{-\infty}^0 \mathrm{d}\hat{\tau}\frac{\exp\{-(\bar{\tau}^2+2\rho\bar{\tau}\hat{\tau}+\hat{\tau}^2)/2(1-\rho^2)\}}{2\pi\sqrt{1-\rho^2}},$$

que por meio da mudança de variáveis

$$\begin{cases} \bar{\tau} &= r\cos\theta \\ \hat{\tau} &= r\,\mathrm{sen}\theta \end{cases} \quad \text{com} \quad \begin{cases} 0 \leqslant r \leqslant \infty \\ -\pi/2 \leqslant \theta \leqslant 0 \end{cases} \tag{A.17}$$

transforma-se em

$$\mathcal{E}_g = \frac{1}{\pi\sqrt{1-\rho^2}}\int_{-\pi/2}^0 \mathrm{d}\theta\int_0^\infty \mathrm{d}r\; r\exp\{-r^2(1+2\rho\,\mathrm{sen}\theta\cos\theta)/2(1-\rho^2)\}.$$

Logo,

$$\begin{aligned} \mathcal{E}_g &= \frac{1}{\pi}\int_{-\pi/2}^0 \mathrm{d}\theta\frac{\sqrt{1-\rho^2}}{1+\rho\,\mathrm{sen}(2\theta)} \\ &= \frac{\sqrt{1-\rho^2}}{\pi}\int_{-\pi/2}^0 \frac{\mathrm{d}\theta}{1+2\rho\tan\theta\cos^2\theta} \\ &= \frac{\sqrt{1-\rho^2}}{\pi}\int_{-\pi/2}^0 \frac{\sec^2\theta\mathrm{d}\theta}{\sec^2\theta+2\rho\tan\theta}. \end{aligned}$$

Fazendo mais uma mudança de variável $u = \tan\theta$

$$\mathcal{E}_g = \frac{1}{\pi}\int_{-\infty}^0 \frac{\sqrt{1-\rho^2}\mathrm{d}u}{(1-\rho^2)+(u+\rho)^2}$$

e por fim $x = (u+\rho)/\sqrt{(1-\rho^2)}$

$$\mathcal{E}_g = \frac{1}{\pi}\int_{-\infty}^{\frac{\rho}{\sqrt{1-\rho^2}}} \frac{\mathrm{d}x}{1+x^2} = \frac{1}{\pi}\left[\arctan(x)\right]_{-\infty}^{\frac{\rho}{\sqrt{1-\rho^2}}} = \frac{1}{\pi}\arctan\left(\sqrt{\frac{1-\rho^2}{\rho}}\right).$$

Portanto, finalmente chegamos a expressão desejada:

$$\mathcal{E}_g = \frac{1}{\pi}\arccos\rho. \tag{A.18}$$

[3]Isso pode ser claramente percebido reescrevendo A.10 na forma $P(\hat{\tau},\bar{\tau}) = e^{-(\bar{\tau}^2+2\rho\bar{\tau}\hat{\tau}+\hat{\tau}^2)/2(1-\rho^2)}/2\pi\sqrt{1-\rho^2}$.

B

O algoritmo ótimo para o perceptron.

O algoritmo ótimo para o perceptron, proposto em [Kinouchi et al 1992], tem como motivação responder qual é a melhor maneira de processar a informação com o intuito de maximizar a capacidade de generalização. Assim, dado um algoritmo hebbiano

$$\hat{\omega}_{\mu+1} = \hat{\omega}_{\mu} + \frac{1}{N}\sigma_{\mu+1}W_{\mu+1}\boldsymbol{\xi}_{\mu+1}, \tag{B.1}$$

procura-se um $W_{\mu+1}$, uma função que pode depender dos campos sinápticos e denominada *função de modulação*, que maximize o decaimento do erro de generalização por dado, ou seja, que maximize:

$$\frac{\mathrm{d}e_g}{\mathrm{d}\alpha} = \frac{\partial e_g}{\partial \rho}\frac{\mathrm{d}\rho}{\mathrm{d}\alpha}. \tag{B.2}$$

Usando os campos A.8 e distribuição A.10 definidos no apêndice anterior e seguindo o mesmo procedimento utilizado na seção 2.2.2, encontramos:

$$\frac{\mathrm{d}\rho}{\mathrm{d}\alpha} = \frac{1}{\hat{\omega}}\left\langle(\bar{\tau} - \rho\hat{\tau})\sigma W - \frac{\rho}{2\hat{\omega}}W^2\right\rangle_{\sigma,\hat{\tau},\bar{\tau}}, \tag{B.3}$$

sendo $\hat{\omega} = \sqrt{\hat{\boldsymbol{\omega}}\cdot\hat{\boldsymbol{\omega}}}$. Com isso, podemos fazer a variação

$$\frac{\delta}{\delta W}\left(\frac{\mathrm{d}e_g}{\mathrm{d}\alpha}\right)_{W=W^*} = 0, \tag{B.4}$$

e encontrar W^*:

$$W^* = \hat{\omega}\left(\sigma\frac{\bar{\tau}}{\rho} - \sigma\hat{\tau}\right). \tag{B.5}$$

Como o módulo do campo $\bar{\tau}$ não é acessível para o perceptron aprendiz, mas apenas $\sigma = \text{sinal}(\bar{\tau})$, o substituímos por

$$\tilde{W} = \int \mathrm{d}|\bar{\tau}| P(\hat{\tau}, \sigma|\bar{\tau}|) W^* \Big/ \int \mathrm{d}|\bar{\tau}| P(\hat{\tau}, \sigma|\bar{\tau}|), \tag{B.6}$$

que resulta na função:

$$\tilde{W} = \hat{\omega}\sqrt{\frac{2r}{\pi}}\frac{\exp\{-\hat{\tau}^2/2r\}}{\operatorname{erfc}(-\sigma\hat{\tau}/\sqrt{2r})}, \tag{B.7}$$

onde definimos $r = (1-\rho^2)/\rho^2$.

Substituindo a expressão B.7 em B.1, encontramos o algoritmo ótimo:

$$\hat{\boldsymbol{\omega}}_{\mu+1} = \hat{\boldsymbol{\omega}}_{\mu} + \frac{1}{N}\hat{\omega}_{\mu}\sqrt{\frac{2r_{\mu}}{\pi}}\frac{\exp\{-\hat{\tau}_{\mu}^2/2r_{\mu}\}}{\operatorname{erfc}(-\sigma\hat{\tau}_{\mu}/\sqrt{2r_{\mu}})}\sigma_{\mu+1}\boldsymbol{\xi}_{\mu+1} \tag{B.8}$$

C

Nota sobre as simulações realizadas.

Todas as curvas apresentadas nessa tese foram obtidas por meio de uma média de 100 simulações com a dimensão $N = 100$ para os vetores sinápticos ($\boldsymbol{\omega}$) e de entrada ($\boldsymbol{\xi}$), exceto quando explicitamente declarado na legenda das figuras — por exemplo, a curva $f(\alpha) = 0,88/\alpha$ nas figuras 2.5, 2.6 e 4.3.

Referências Bibliográficas

[Amaral et al 2004] Selene R. Amaral, Said R. Rabbani and Nestor Caticha, *Multigrid priors for a Bayesian approach to fMRI.* Neuroimage, Vol. 23, 435-775 (2004).

[Amari et al 1998] S. Amari and S.C. Douglas, *Why natural gradient?* Proc. Int. Conf. Acoustic., Speech, Signal Processing, vol II, 1213-1216. Seattle, WA (1998).

[Amari 1998] S. Amari, *Natural gradient works efficiently in learning.* Neural Networks 10, 251-276 (1998).

[Andrieu et al 2003] C. Andrieu, N. de Freitas, A. Doucet and M. I. Jordan, *An introduction to MCMC for machine learning.* Machine Learning 50, 5-43 (2003).

[Bernado et al 2000] José M. Bernado and Adrian F.M. Smith, *Bayesian theory.* John Wiley & Sons (2000).

[Caticha et al 2004] Ariel Caticha and Roland Preuss, *Maximum entropy and Bayesian data analysis: Entropic and prior distribution.* Phys. Rev. E Vol 70, 046127 (2004).

[Caticha et al 2001] Nestor Caticha and Evaldo A. de Oliveira, *Gradient descent learning in and out of equilibrium.* Phys. Rev. E. Vol 63, 061905 (2001).

[Challet et al 1997] D. Challet and Y. Zhang, *Emergence of cooperation and organization in an evolutionary game.* Physica A, vol. 246, 407-418 (1997).

[Copelli et all 1997] M. Copelli, R. Eichhorn, O. Kinouchi, M. Biehl, R. Simonetti, P. Riegler and N. Caticha, *Noise robustness in multilayer neural networks.* Europhys. Lett., vol 37(6), 427-432 (1997).

[Cox 1946] R.T. Cox, *Probability, frequency and reasonable expectation.* American Journal of Physics. Vol. 14, 1-13 (1946).

[Duda et al 2001] Richard O. Duda, Peter E. Hart and David G. Stork, *Pattern classification.* John Wiley & Sons (2001).

[Engel et al 2001] Andreas Engel and C. Van den Broeck, *Statistical mechanics of learning.* Cambridge Univ. Press (2001).

[Finger 1994] Stanley Finger, *Origins of neuroscience, a history of explorations into brain function.* Oxford University Press (1994).

[Haykin 2003] Simon Haykin, *Redes Neurais - Princípios e prática.* Artmed Editora LTDA (2003).

[Gama et al 2004] J. Gama, Pedro Medas, G. Castillo and P. Rodrigues, *Learning with drift detection.* Ana L. C. Bazzan, Sofiane Labidi (Eds.): Advances in Artificial Intelligence - SBIA 2004, 17th Brazilian Symposium on Artificial Intelligence, São Luis, Maranhão, Brazil. Lecture Notes in Computer Science, Springer Verlag 3171, 285-295 (2004).

[Jaynes 1982] E.T. Jaynes, *On the rationale of maximum-entropy methods.* Proceedings of the IEEE, vol. 70, n. 9, 939-952.

[Jaynes 1988] E.T. Jaynes, *How does the brain do plausive reasoning?* Em Maximum-Entropy and Bayesian Methods in Science and Engineering. Vol. 1, 1-24 (1988).

[Jaynes 2003] E.T. Jaynes, *Probability theory : the logic of science.* Ed. G. L. Bretthorst. Cambridge University Press (2003).

[Jefferys et al 1992] William H. Jefferys and James O. Berger, *Ockham's razor and bayesian analysis.* Am. Scient. Vol. 80, 64-72 (1992).

[Kinouchi et al 1992] O. Kinouchi and N. Caticha, *Optimal generalization in perceptrons.* J. Phys. A. Vol 25, 6243-6250 (1992).

[Kinouchi et al 1996] O. Kinouchi and N. Caticha, *Learning algorithms that gives the Bayes generalization limit for perceptrons.* Phys. Rev E. Vol 54, R54 (1996).

[Kirsch 1996] A. Kirsch, *An introduction to the mathematical theory of inverse problems.* Spring-Verlag (1996).

[Koychev 2000] I. Koychev, *Gradual Forgetting for Adaptation to Concept Drift.* In Proceedings of ECAI 2000 Workshop Current Issues in Spatio-Temporal Reasoning, Berlin (2000).

[Levin et al 1990] E. Levin, N. Tishby and S. Solla, *A statistical approach to learning and generalization in layered neural networks.* Proc. IEEE, Vol 78, 1568-1574 (1990).

[Milner 1963] B. Milner, *Effects of differente brain lesions on card sorting — role of frontal lobes.* Archives of neurology 9, 110-11 (1963).

[Murata et al 2002] M. Murata, M. Kawanabe, A. Ziehe, K. Müller and S. Amari, *On-line learning in changing environments with applications in supervised and unsuperviside learning.* Neural Networks 15, 743-760 (2002).

[de Oliveira 2000] Evaldo A. de Oliveira Filho, *Relações entre aprendizagem dentro e fora de equilíbrio termodinâmico.* Dissertação de mestrado apresentada ao Instituto de Física da Universidade de São Paulo (2000).

[Opper et al 1990] M. Opper and D. Haussler, *Generalization performance of Bayes optimal classification algorithm for learning a perceptron.* Phys. Rev. Lett. 66, 2677-2680 (1990).

[Opper 1998] M. Opper, *A bayesian approach to on-line learning.* Em On-line Learning in Neural networks. Ed. D. Saad, Cambridge Univ. Press, 363-378 (1998).

[Reents et al 1998] G. Reents and R. Urbanczik, *Self-averaging and on-line learning.* Phys. Rev. Lett. 80, 5445-5447 (1998).

[Rosenblatt 1958] F. Rosenblatt, *The perceptron: a probabilistic model for information storage and organization in the brain.* Cornel Aeronautical Laboratory, Psychological Review, vol. 65, 386-408 (1958).

[Scholkopf et al 2001] Bernhard Scholkopf and Alexander J. Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond.* The MIT Press (2001).

[Sivia 2000] D. S. Sivia, *Data analysis - a bayesian tutorial.* Oxford Univ. Press (2000).

[Solla et al 1998] Sara A. Solla and Ole Winther, *Optimal perceptron learning: an online bayesian approach.* Em On-line Learning in Neural networks. Ed. D. Saad, Cambridge Univ. Press, 379-398 (1998).

[Tikhonov et al 1977] A. N. Tikhonov and V.Y. Arsenin, *Solutions of ill-posed problem*, Winston, New York, 1977

[Vapnik 1998] Vlamimir N. Vapnik, *Statistical learning theory.* Wiley, John & Sons (1998).

[Vicente et al 1998] Renato Vicente, Osame Kinouchi and Nestor Caticha, *Statistical mechanics of online learning of drifting concepts: a variational approach.* Machine Learning 32, 179-201 (1998).

[Walker 1969] A. M. Walker, *On the asymptotic behaviour of posterior distributions.* J. R. Statist. Soc. B 30, 80-88 (1969).

[Widmer et al 1996] G. Widmer and M. Kubat, *Learning in the presence of concept drift and hidden contexts.* Machine Learning 23, 69-11 (1996).

Índice Remissivo

Então Pedro começou a falar: "Agora percebo verdadeiramente que Deus não trata as pessoas com parcialidade, mas de todas as nações aceita todo aquele que o teme e faz o que é justo. Vocês conhecem a mensagem enviada por Deus ao povo de Israel, que fala das boas novas de paz por meio de Jesus Cristo, Senhor de todos. Sabem o que aconteceu em toda a Judéia, começando na Galiléia, depois do batismo que João pregou, como Deus ungiu a Jesus de Nazaré com o Espírito Santo e poder; e como ele andou por toda parte fazendo o bem e curando todos os oprimidos pelo diabo, porque Deus estava com ele.

Nós somos testemunhas de tudo o que ele fez na terra dos Judeus e em Jerusalém. A este mataram, suspendendo-o num madeiro. Deus, porém, o ressuscitou no terceiro dia e fez que ele fosse visto, não por todo o povo, mas por testemunhas que designara de antemão, por nós que comemos e bebemos com ele depois que ressuscitou dos mortos. Ele nos mandou pregar ao povo e testemunhar que este é aquele a quem Deus constituiu juiz de vivos e mortos. Todos os profetas dão testemunho dele, de que todo aquele nele crê recebe o perdão dos pecados mediante o seu nome".

Atos 10:34-43.